SYS·CON

NOW INCORPORATING XML JOURNAL

Web Services JOURNAL

.NET J2EE XML

June 2005 Volume 5 Issue 6

Managing and Integrating Extended Supply Chains

Web services to the rescue

pg. 16

Eight Things SOA Is Not

What not to do in your next SOA rollout

pg. 22

Managing Enterprise Data Complexity Using Web Services: Part 1

Data services architecture

pg. 28

Two To TANGO

SOA and Business Process Management

pg. 10

RETAILERS PLEASE DISPLAY UNTIL AUGUST 31, 2005

$6.99US $7.99CAN

06

71486 03420 9

Parasoft SOAPtest®

Keep Your Skills Ahead of the Crowd

Keeping your IT skills ahead of the crowd is not as difficult as most people fear. Staying on top of the trends may seem like a daunting task if, like most people, you assume that each new technology is a completely new invention that you must learn from the ground up. Fortunately, nothing is really all that new. Inventors typically create new technologies by studying existing technologies, then building upon them in ways that extend and improve them. 100% new technological advancements are very rare.

Inventors almost always leverage legacy technologies as they invent new ones. Why not leverage your own knowledge of those legacy technologies as you try to learn about the new inventions? To learn about new technologies as painlessly as possible, consider how each new advancement is similar to what you already know.

For example, consider Web services. Web services are a new trend, but — at a technological level — the parts of a Web service are not all that unique. Web services are based on remote procedural calls — messages sent to a server, which calls the requested function. RPCs were developed years ago, and are hardly a new concept. Really, the only "new" thing in Web services is the standard that is being used to write the application. If you break down Web services in this way, it's easy to learn about them. To continue with this process, you might next explore the payload requirements, the process for determining what function to call, and how the call works. As you can imagine, it's a lot more efficient — and interesting — to learn about a new technology based on its relation to familiar technologies than to learn about it by reading the specification cover to cover.

As always, the devil is in the details. But most details are critical only if you want to specialize in a given technology. For instance, if you want to specialize in Web services, you need to familiarize yourself with the details of Web service development. In that case, your next step would be to learn how to format the messages, how to expose Web services, and so on.

*– **Adam Kolawa,** Ph.D.*
Chairman/CEO of Parasoft

It's automated. It's fast. And it's the most versatile Web Services testing tool.

SOAPtest® enables you to deliver better Web Services in less time.

The simple fact is, no other Web service testing product can do what SOAPtest can do. From verifying functionality and performance to ensuring security and interoperability, SOAPtest automates all critical testing processes across the entire lifecycle of your Web service.

But don't take our word for it.... Go to www.parasoft.com/soaptest_WSD and try it for free — no commitment, no obligation. If you like it (and we suspect you will) just let us know. We'll be more than happy to help you and your development team get up and running.

For Downloads go to www.parasoft.com/soaptest_WSD PARASOFT®

Email: soaptest@parasoft.com Call: 888-305-0041 x1209

Features

- WSDL schema verification and compliance to standards
- Automatic test creation using WSDL and HTTP Traffic
- Data-driven testing through data sources (Excel, CSV, Database Queries, etc)
- Scenario-based testing through XML Data Bank and Test Suite Logic
- Flexible scripting with Java, JavaScript, Python
- WS-I Conformance: Basic Profile 1.0
- WS-Security, SAML, Username Token, X.509, XML Encryption, and XML Signature support
- WS-Security Interop testing emulator
- MIME Attachment support
- Asynchronous Testing: JMS, Parlay (X), SCP, and WS-Addressing support
- Windows Perfmon, SNMP, and JMX monitors
- Detailed Report generation in HTML, XML and Text formats
- Real-Time graphs and charts

Benefits

- Uniform test suites can be rolled over from unit testing to functional testing to load testing
- Prevent errors, pinpoint weaknesses, and stress test long before deployment
- Ensure the reliability, quality, security and interoperability of your Web service
- Verify data integrity and server/client functionality
- Identify server capabilities under stress and load
- Accelerate time to market

Protocol Support

- HTTP 1.0
- HTTP 1.1 w/Keep-Alive Connection
- HTTPS
- TCP/IP
- JMS

Platforms

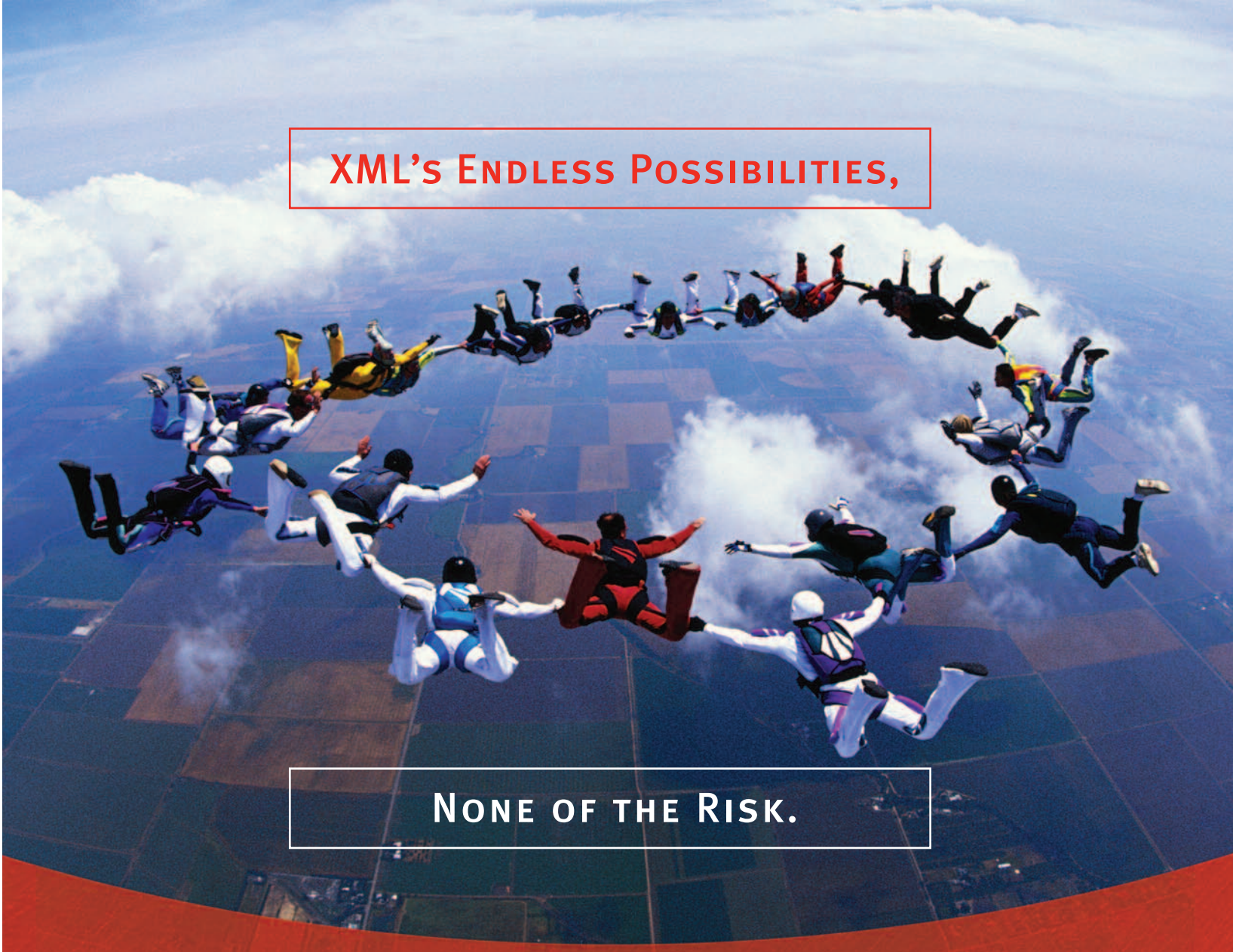
- Windows 2000/XP
- Linux
- Solaris

Contact Info:

Parasoft Corporation
101 E. Huntington Dr., 2nd Flr.
Monrovia, CA 91016

www.parasoft.com

XML's Endless Possibilities,



None of the Risk.

Forum XWall™ Web Services Firewall - Reinventing Security

Security should never be an inhibitor to new opportunity: Forum XWall™ Web Services Firewall has been enabling Fortune 1000 companies to move forward with XML Web services confidently. Forum XWall regulates the flow of XML data, prevents unwanted intrusions and controls access to critical Web services.

Visit us at www.forumsys.com to learn more about how you can take your next leap forward without increasing the risks to your business.

FORUM SYSTEMS™ – THE LEADER IN WEB SERVICES SECURITY

WebServices JOURNAL
.NET J2EE XML

June 2005 Volume 5 Issue 6

XML JOURNAL
Inside WSJ

10 Service-Oriented Architecture and Business Process Management

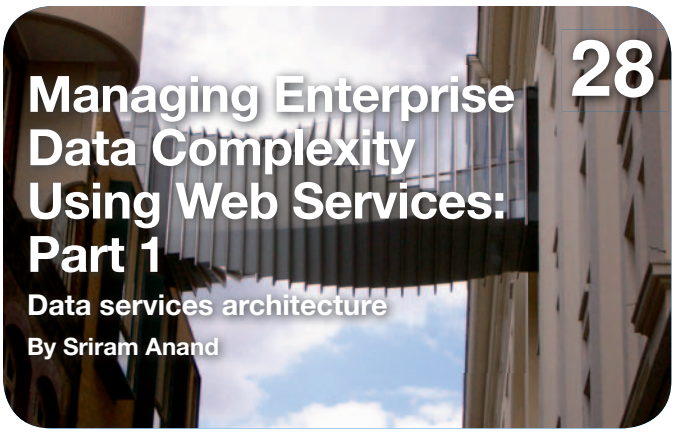
Two to Tango

By Deepak Pareek

22 Eight Things SOA Is Not

What not to do in your next SOA rollout

By Paul O'Connor



28 Managing Enterprise Data Complexity Using Web Services: Part 1

Data services architecture

By Sriram Anand

WebSphere.

Key

1. Deductible viewed quickly.
2. Claim info filed accurately.
3. Vendor receives complete data.
4. Quotes researched easily.
5. Great service boosts policy sales.

MIDDLEWARE IS IBM SOFTWARE. Powerful WebSphere software. For an integrated IT environment that makes your business more flexible. WebSphere delivers a secure, scalable platform. Enabling a service-oriented architecture. So your business can respond faster. More efficiently. To partners, vendors and customers. With no ripping. No replacing. No headaches. No kidding.

Middleware for the on demand world. Learn more at **ibm.com**/middleware/flexible ON DEMAND BUSINESS™

Sun microsystems

Java™

INNOVATE
with the Power of Java™

Connect with the full power of Java™ technology at the JavaOne℠ conference.

In-depth EDUCATION
Evolve your skills in 300+ of expert-led technical and Bof sessions.

Real-world INNOVATION
Evaluate proven tools and technologies in the JavaOne℠ Pavilion.

Global COMMUNITY
Celebrate the tenth anniversary of Java technology.

Visionary INSIGHT
Hear what the future holds from industry leaders.

June 27–30, 2005

JavaOne℠ Pavilion: June 27–29, 2005
Moscone Center, San Francisco, CA

Experience a week unlike any other

EXPERIENCE THE 10TH ANNUAL JAVAONE℠ CONFERENCE.

Core Platform | Core Enterprise | Desktop | Web Tier | Tools | Mobility and Devices | Cool Stuff

Save $100!

REGISTER
by June 27, 2005 at java.sun.com/javaone/sf.

JavaOne℠
Sun's 2005 Worldwide Java Developer Conference℠

WebServices JOURNAL
.NET J2EE XML

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending

Montvale, NJ 07645 and additional mailing offices
POSTMASTER: Send address changes to:
WEB SERVICES JOURNAL, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

Elementary Education

WRITTEN BY
SEAN RHODY

When I was in college, I considered becoming a teacher, but then the thought of the pay scale as well as some time spent substitute teaching convinced me that computers might be a better way to go. I mention this because I never did get the chance to take any education courses, so I don't know if the way we teach the craft of software development is wrong, or just seems that way because I'm not fully aware of all of the issues. If this seems a strange way to begin approaching the concept of enterprise architecture, stay with me for a moment.

When I studied Computer Science, the curriculum began with a basic language course, then went into some course work on computer and number theory, and later went to databases and operating systems. While fully half of my degree requirements were spent on computers, there still wasn't any requirement for course work in software engineering or architecture. To me, this seems as though it should be the first requirement – give the student a good overview and grounding on all of the concepts involved in developing software, not just courses in languages.

A career in computers usually mirrors that trend. You start as a programmer, building small pieces of a puzzle, maybe some shell scripts, some utilities, pieces of an application here and there. Later you may own an entire application, which you may or may not have built from scratch. Along the way you get exposed to other concepts such as QA, deployment, and shared services. After years of this, you become an architect, and add the design process to your bag of tricks.

Architecture can mean many things to many people. A technical architect may deal with infrastructure, an application architect may design applications, and an enterprise architect typically leads a sweeping effort involving a corporate application and infrastructure portfolio. A program architect may be a leader of architecture teams implementing applications using an overall architecture; essentially he is a practical, or practicing, enterprise architect.

In an enterprise there is a strong need to communicate enterprise-level concepts down to application architects and developers. This includes standards, frameworks, and any other decision points that can affect a software designer or developer. Security is a good example. Without knowing that an enterprise security system, such as LDAP, is in place, an application designer or developer might design a proprietary, database-based solution for application security. Knowing that LDAP was to have been used could have saved development time in the first place, as well as removing the need for rework once the application is rolled out. It might even save time if the application had to be delayed based on the omission.

More important, however, an enterprise architecture contributes to business agility and returns value to the bottom line. With such an architecture, the point solutions and silos become sources of services that can move more rapidly to accommodate changing business conditions. With today's market conditions, regulatory environment, and privacy rules, it is more important than ever that enterprises be able to move rapidly to change the way they do business.

Also, this has a direct impact on the bottom line. Several financial services companies were recently assessed fines in the tens of millions of dollars for failing to produce required annual documentation. Avoiding a 20 million dollar penalty might be enough to finance an entire renovation program. So you can see that enterprise architecture isn't just about solving IT problems; in order to effectively address architecture strong communication is required.

Reuse is a common IT catchphrase. Services represent the ultimate in reuse but rigor must be enforced to ensure that a catalog of services is correct and is not adulterated. One off, "just for my division" special versions of services lead to organizational nightmares. It is just as much the architect's job to oppose new systems as it is to propose them. Duplicate systems are sometimes necessary – you cannot boil the ocean – but redundant functionality should be opposed at every turn. Using redundancy to avoid a service outage makes sense, but letting it stop your business from doing business because of extra systems doesn't.

So back to the basics – you need an enterprise architecture. And you need to make sure people get the high-level overview, before they dive down into the details. Just like at school, I wish someone had told me this at the start.

About the Author

Sean Rhody is the editor-in-chief of *Web Services Journal*. He is a respected industry expert and a consultant with a leading consulting services company.

sean@sys-con.com

WebServices JOURNAL .NET J2EE XML

PRESIDENT AND CEO
Fuat Kircaali fuat@sys-con.com
VP, BUSINESS DEVELOPMENT
Grisha Davida grisha@sys-con.com
GROUP PUBLISHER
Jeremy Geelan jeremy@sys-con.com

ADVERTISING
SENIOR VP, SALES & MARKETING
Carmen Gonzalez carmen@sys-con.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
NATIONAL SALES & MARKETING MANAGER
Dennis Leavey dennis@sys-con.com
ADVERTISING MANAGER
Megan Mussa megan@sys-con.com
ASSOCIATE SALES MANAGERS
Dorothy Gil dorothy@sys-con.com
Kim Hughes Kim@sys-con.com

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
NATIONAL SALES MANAGER
Jim Hanchrow jimh@sys-con.com

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com

sys-con.com
CONSULTANT, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Percy Yip percy@sys-con.com
Vincent Santaiti vincent@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com
ACCOUNTS RECEIVABLE
Gail Naples gailn@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-201-802-3012
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
Newsstand Distribution Consultant:
Brian J. Gregory / Gregory Associates / W.R.D.S.
732 607-9941 - BJGAssociates@cs.com

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

SYS-CON MEDIA

Distributing Excellence

WRITTEN BY
AJIT SAGAR

As SOA and Web services adoption in the industry is gaining more momentum, the need to get quick wins and to show the value of adopting new (or old) paradigms is weighed against the risk of facing the repercussions of slapping something together in a quick and dirty fashion and paying the higher cost later. Many of our smart clients (not to be confused with .NET smart clients) are putting together the right groups to facilitate the adoption of these new technologies across their organizations.

The deployment of SOA is centered on governance. In order to have an efficient governance process, central groups that can act as COEs (Centers of Excellence) need to be in place before portfolios are allowed to develop and deploy functionality for their respective applications. There are too many technologies, business processes, vendor products, and confusing messages in the mix for any organization to leverage the benefits of adopting SOA through Web services effectively. Research needs to be conducted, products need to be evaluated, eliminated, and selected, and patterns and guidelines need to be published to address the common needs of the applications. This obviously has to be balanced against the drivers from business to show value ASAP.

Obviously cost is the main hindrance. It is very hard for a COE to show tangible cost savings. Unfortunately, the software industry hasn't developed appropriate metrics to assess the value provided by such groups. Fortunately, factors such as unfamiliarity with new technology and lack of required skillsets in individual application groups drives the need for establishing centers of excellence. It is still a hard sell to maintain such bodies, which are viewed by business owners as cost centers.

SOA lends itself very well to the outsourcing and off shoring model. An interesting side effect of this development is that a center of excellence can be distributed across geographies to manage the economies of scale. There is no getting away from the fact that the ultimate distribution and sharing of knowledge needs to be at the location where the applications are being developed. But this does not mean that the work cannot be distributed.

While this is an attractive proposition, it requires careful planning to set it up. To set up an effective distributed COE, appropriate roles need to be defined and staffed at the client site, as well as at the offshore/nearshore locations. Effective project management and processes need to be set up to produce deliverables such as white papers, vendor evaluations, newsletters, FAQs, blueprints, architecture cookbooks, patterns and guidelines documents, etc. Appropriate infrastructure needs to be set up to enable effective communication and knowledge sharing.

As of yet, this is not a model that many organizations have thought through or established. Fortunately, in our client engagements, we have seen traction from several clients and genuine interest in taking advantage of the distributed nature of large global consulting shops. At Infosys we have been able to offer attractive value propositions for setting up such organizations. One of the primary requirements to be able to effectively deliver on promises made in such initiatives is to have a well-established research facility and alliances with the appropriate vendors in the SOA and Web services space. The distributed COE model that we have deployed is illustrated in Figure 1. The diagram illustrates how a center of excellence for Web services and SOA is established for a U.S. client, with portfolios distributed across different states. The states shown in the diagram have been changed from the actual sites where the COE has been implemented to maintain client confidentiality.

The bottom line is that if you have a large sized

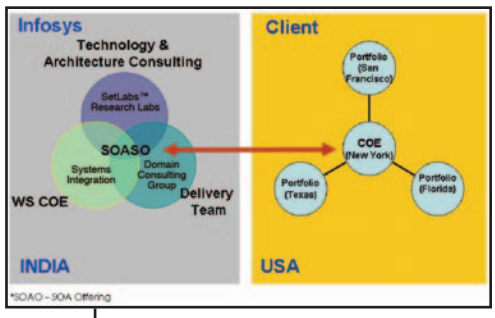


FIGURE 1 **Distributed COE for SOA and Web services**

IT with multiple portfolios, and if you don't consider standardization and governance across your multitude of applications, then the cost that will be paid in the long run offsets the savings that you will achieve in the immediate future. Don't be penny-wise and pound-foolish.

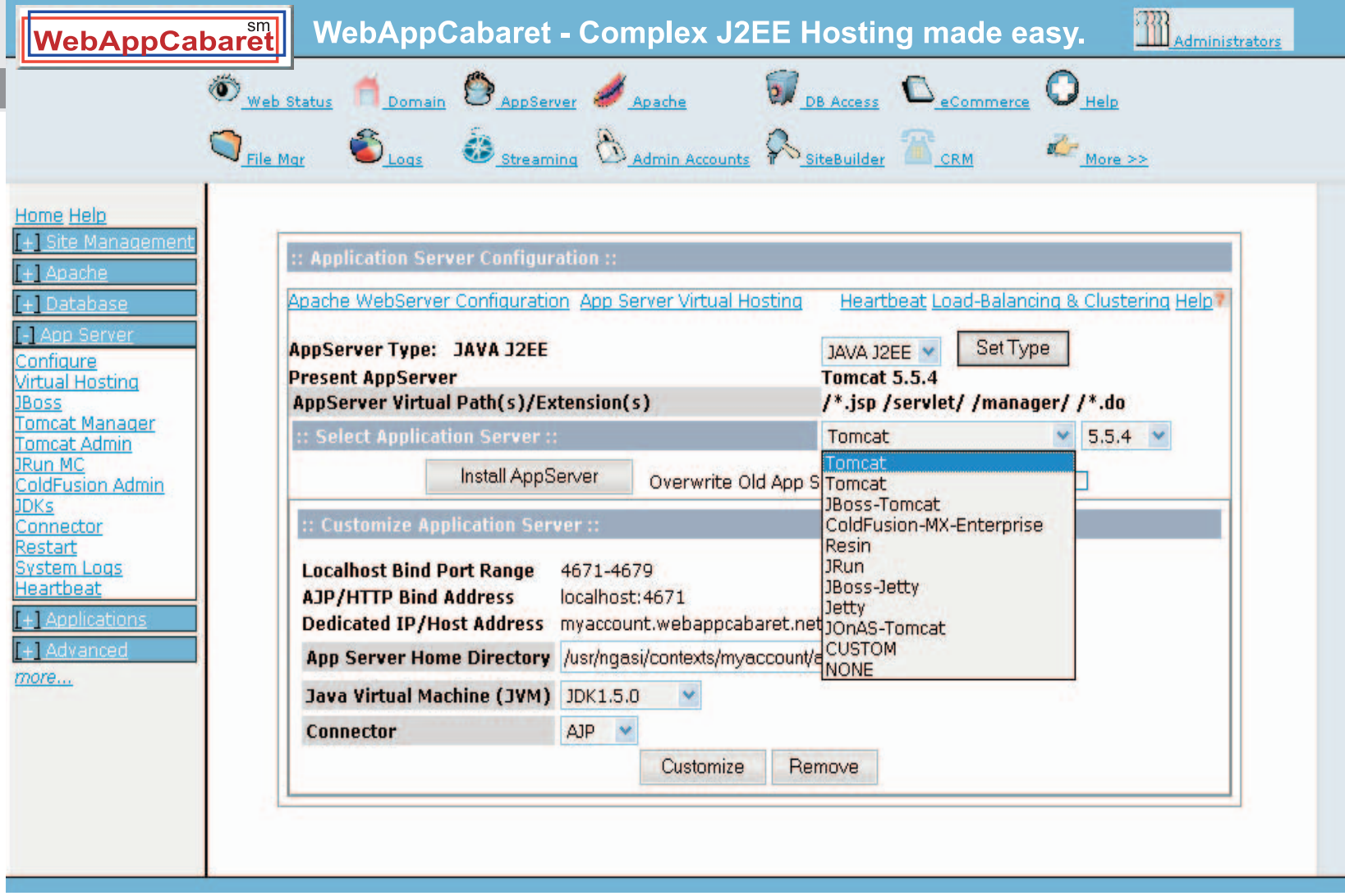
About the Author

Ajit Sagar is a senior technical architect with Infosys Technologies, Ltd., a global consulting and IT services company. Ajit has been working with Java since 1997, and has more than 15 years experience in the IT industry. Ajit is *JDJ*'s Enterprise editor, the founding editor of *XML Journal*, and has been a frequent speaker at SYS-CON's Web Services Edge series of conferences.

ajitsagar@sys-con.com

WebAppCabaret sm

WebAppCabaret - Complex J2EE Hosting made easy.



Imagine a hosting company dedicated to meet the requirements for complex web sites and applications such as those developed with Java J2EE.
At WebAppCabaret our standards based process and tools make deploying Java J2EE applications as easy as a point-and-click.
We call it **Point-and-Deploy Hosting**.
Our advanced NGASI Web Hosting management Control was designed for the hosting and management of complex web sites and applications thus cutting down on maintenance time.

Backed by an experienced staff as well as a **Tier 1 Data center** and network. Our network is certified with frequent security audits by reputable security firms.

All hosting plans come with **advanced tools** to manage your application server and Apache web server. Not to mention the other features, such as virus-protected email, bug tracking, and many more portal components.

Complete power and control at the tip of your fingers. We take care of the system and hosting infrastructure so you can concentrate on development and deployment of your application. That is **real ROI**.

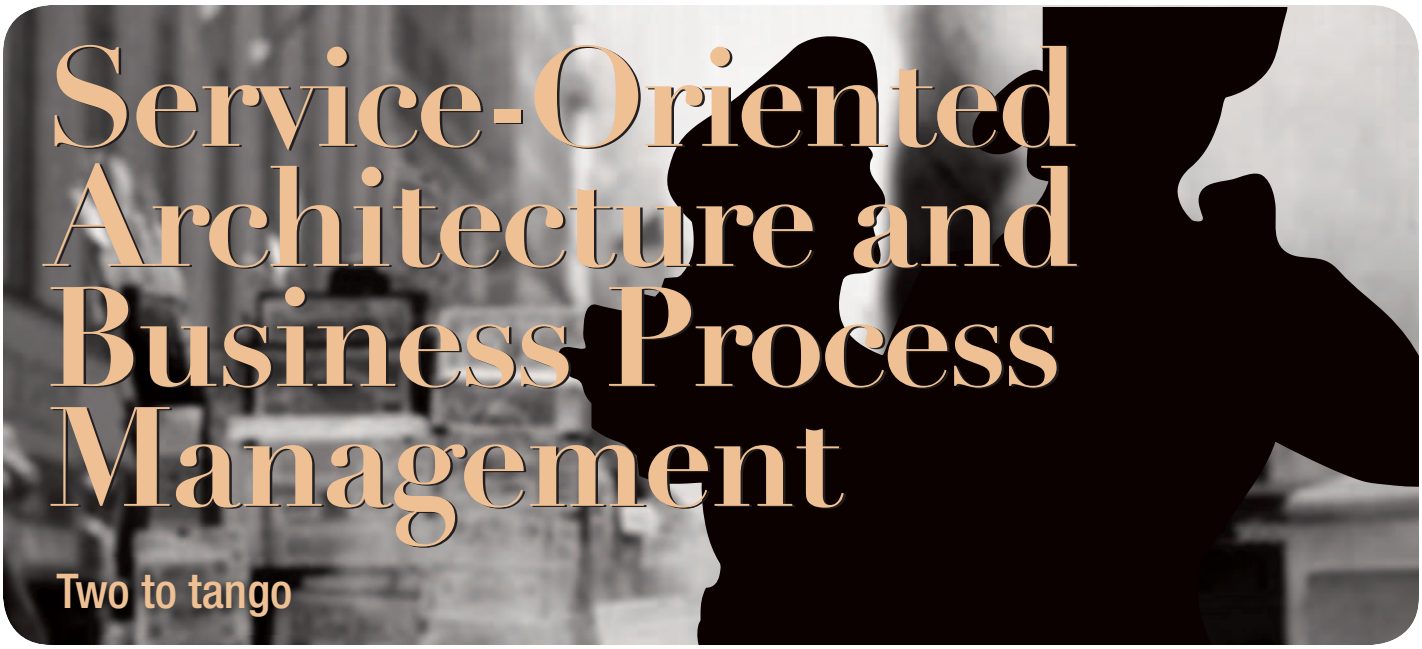
Log on now at http://www.webappcabaret.com/jdj.jsp or call today at 1.866.256.7973

NGASI POWERED

WebAppCabaret sm

http://www.webappcabaret.com/jdj.jsp 1.866.256.7973

Point-and-Deploy J2EE Hosting



Service-Oriented Architecture and Business Process Management

Two to tango

Today's business environment is changing rapidly. Business dynamics and technological innovations have left organizations with a disparate mix of operating systems, applications, and databases – making it difficult, time consuming, and costly for IT departments to deliver new applications that integrate heterogeneous technologies. The key to success in the networked economy is not only being able to create business processes to automate value chains, but also being able to modify these processes as business requirements change. Innovation in enterprise architecture will come from service-oriented architecture (SOA) and business process management (BPM).

WRITTEN BY
DEEPAK PAREEK

Enterprises must rethink their IT infrastructure and begin to implement an SOA and BPM. The combination of BPM and an SOA are crucial steps towards becoming a real-time enterprise, thus creating the foundation to respond faster to changing business requirements and to react to events in real time. Advances in Web services technology and standards are critical driving forces that will take SOA and BPM to mainstream adoption.

This article examines concepts of SOA and BPM, evaluates the benefits of these concepts individually, and articulates expectations for them. Further, it proposes and explains the SOA-BPM framework and its components. Finally this article makes a case about synergy between SOA and BPM and how each one can enhance the returns of the other.

Until now, SOAs have only been implemented by a few leading-edge enterprises due to high costs and the level of technical skill required; however, Web services are now making it both affordable and possible from a skills point of view. In addition, the consolidation of security technologies and maturing standards for Web services means security is no longer a stumbling block. This will accelerate the spread of SOA and make it mainstream. Web services used to be a solution looking for a problem. With SOA, it has finally found it.

BPM improves process design and integration, thereby making application systems work more efficiently together. It delivers tactical cost/time benefits, while building a base for competitive growth. Regulatory compliances such as Sarbanes-Oxley or Basel 2 are additional drivers for BPM, as they require the monitoring of critical business processes and the ability to report abnormal situations in the processes themselves.

SOA allows companies to reuse existing applications and data to create new business processes. It makes the enterprise more agile and less locked-in to certain ways of doing business, as applications can be changed faster and more easily. The impending risk of standard fragmentation will make it more complex for enterprises to reap the full benefits of SOA and BPM during the next three years. However, enterprises must invest now to create the skills and governance processes necessary to leverage SOA and BPM for business advantage.

Enterprises have traditionally implemented separate solutions for operating legacy and packaged applications, business-to-business (B2B) interactions, collaboration, and general-

Global Knowledge™
Experts Teaching Experts

Take our 10-minute assessment.

It's no secret that Red Hat Enterprise Linux is the most widely deployed Enterprise Linux solution in the world today. In fact, Certification Magazine recently named the Red Hat Certified Engineer (RHCE) program the highest quality certification in all of IT. Global Knowledge, a premier partner for Red Hat training, invites you to test your skills and pre-assess for Linux success! **Visit globalknowledge.com/lw for details.**

redhat PREMIER PARTNER

purpose distributed computing. Moreover, IT professionals also need to plan for unforeseen and changing dynamics created by mergers and acquisitions, new partnerships, expansion, and new customer requirements. This creates a serious bottleneck in the ability to manage, change, and modify enterprise processes to dynamically match changes in requirements.

The key to success in the networked economy is the ability to create and modify processes to automate value chains in concert with changing requirements. Faster change management will help enterprises integrate their processes over the Internet so they can achieve greater efficiency, generate more revenue, and enter new markets.

A new category of enterprise infrastructure solutions, built on an SOA, will deliver these benefits. SOAs are based on the notion of services, which are high-level software components that include Web services. Implementation of an SOA requires tools as well as run-time infrastructure software, which we collectively refer to as an SOA implementation framework (SOAIF). An SOAIF includes both design-time and run-time capabilities, as well as all of the software functionality that an enterprise requires to build and operate an SOA, including service-oriented:

- Tools
- Management
- Integration
- Processes

Users compose processes by connecting multiple service instances using visual tools, while the SOAIF provides the run-time deployment infrastructure across the network for running the application and process. An SOAIF lets business analysts create, deploy, manage, and change processes spanning multiple enterprise applications, departments, and partners.

SOAIFs address the needs of process management at the application, human-interaction, and implementation levels. The SOAIF addresses these needs within and across enterprises and across multiple domains, including EAI, B2B integration, BPM, collaboration, and even network management; each area is traditionally served by distinct solutions. BPM is a very important and vital part of enterprise SOA implementation.

What Is Business Process Management?

BPM is the practice of improving the efficiency and effectiveness of any organization by automating the organization's business processes. Some key issues of BPM worth noting are as follows.

- Many companies have business processes that are unique to their business models. Since these processes tend to evolve over time as the business reacts to market conditions, the BPM solution you choose must be easily adaptable to the new conditions and requirements and continue to be a perfect fit for the company.
- In order to use BPM effectively, organizations must stop focusing exclusively on data and data management, and adopt a process-oriented approach that makes no distinction between work done by a human and work done by a computer.
- The idea of BPM is to bring processes, people, and information together.
- Identifying the business processes is relatively easy. Breaking down the barriers between business areas, and finding owners for the processes, is difficult.
- BPM not only involves managing business processes within the enterprise; it also involves the real-time integration of the processes of a company with those of its suppliers, business partners, and customers.
- BPM involves looking at automation horizontally instead of vertically.

Now Let's Evaluate Key Components of BPM

- ***BPM Integrated Design Environment.*** BPM IDE is an integrated design environment used to design processes, rules, events, and exceptions. Creating a structured definition of each process is very important to any business, and the IDE enables a business user to design all of processes without any help from IT.
- ***Process Engine.*** The process engine of a BPM solution keeps track of the states and variables for all of the active processes. Within a complex system, there could be thousands of processes with interlocking records and data.
- ***User Directory.*** Administrators define people in the system by name, department, role, and even by potential authority level. This directory will enable tasks to be sent automatically to the defined resources.
- ***Workflow.*** This is the communication infrastructure that forwards tasks to the appropriate individual.
- ***Reporting/Process Monitoring.*** Enables users to track the performance of their current processes and the performance of personnel

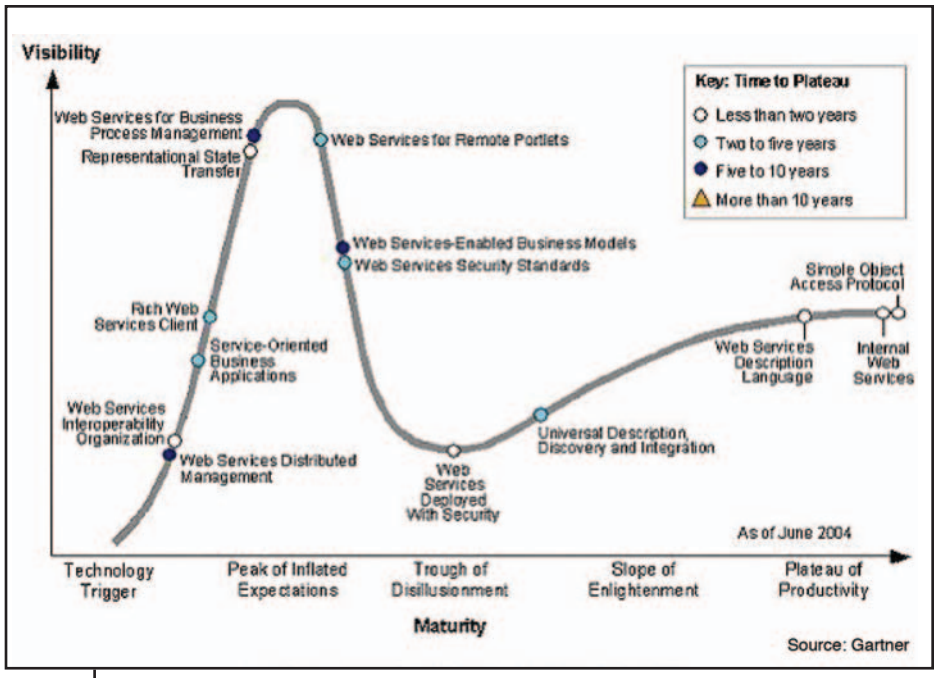


FIGURE 1 | **Technology visibility vs. maturity curve**

who are executing these processes.
- ***Integration.*** Enterprise application integration (EAI) and/or Web services are critical to BPM, as business processes will require data from disparate systems throughout the organization.

What You Should Expect from BPM

- BPM makes it easy for companies to program their current processes, automate their execution, monitor their current performance, and make on-the-fly changes to improve the current processes.
- The process-managed enterprise is the company of the future.
- BPM software enables you to automate those tasks that are currently being performed manually. Many of these tasks require some type of application process, approval or rejection process, notifications, and status reports. A BPM solution can make these processes automatic.
- Exception handling is an area where BPM really shines. Organizations have few problems when their processes run smoothly 99 percent of the time. However, it's the exceptional processes in the 1 percent that dominate the majority of the company's time and resources.
- BPM is excellent for processes that extend beyond the boundaries of an enterprise and communicate with processes of the partners, customers, suppliers, and vendors.

Benefits of an SOA and BPM Framework

An SOA, along with BPM, focuses on internal and cross-enterprise processes, thus helping organizations streamline operations, reduce costs, and increase responsiveness. Specifically, an SOA in conjunction with BPM provides general-purpose, service-based distributed computing capabilities that deliver:

- Faster response rate to changing business requirements
- Operational efficiencies
- Faster, less expensive application integration
- Easier application development and deployment

Responsiveness

Existing enterprise solutions are inadequate in their ability to quickly change processes in response to changing business dynamics. The effort typically requires additional manual code development and results in a system that is difficult to maintain and extend. An SOA and BPM infrastructure strengthens the enterprise by enabling rapid changes to existing processes by dynamically allowing the inclusion of additional services or the modification of existing services. An SOA and BPM framework also includes support for run-time deployment, thereby allowing modified processes to be redeployed instantly.

Efficiency

Most packaged enterprise applications perform well in streamlining processes related to standard tasks. However, the performance rapidly deteriorates while automating and streamlining customized processes that encompass multiple enterprise applications. The process is difficult, time consuming, and expensive to implement and maintain.

The SOA and BPM infrastructure addresses this issue by allowing the definition of any process in any network topology, spanning multiple enterprise boundaries. This is accomplished via a peer-to-peer messaging infrastructure with distributed security mechanisms that allow efficient data exchanges for easy implementation, while enabling each enterprise to enforce its own security policies. This allows an SOA and BPM framework to increase operational efficiency across the entire value chain.

Application Integration

Existing packaged application integration solutions are complex and require significant implementation effort, often including extensive manual coding for deployment purposes. An SOA and BPM infrastructure provides native support for run-time deployment of services across the network and dramatically reduces the overall costs of application integration and deployment by automating these time-consuming processes. It also allows extension of integration across business boundaries.

Application Development and Deployment

In the traditional software development process, translating requirements into working distributed systems is both time consuming and difficult, requiring several stages of manual development and deployment. This complex, error-prone task can be effectively streamlined using a higher-level, component-based SOAIF. The SOAIF incorporates tools that let processes that are developed, using standards such as Business Process Execution Language (BPEL), be easily translated into distributed high-level services, which are easier to develop, manipulate, and debug. These services are easily composed into implementation-level data flows without the user or developer having to track complex middleware concepts, such as topics or queues. Further, the implementation-level services can run on any machine across the network by virtue of the built-in dynamic deployment support SOAIF provides. The combination of service-oriented tools and built-in support for distributed debugging, run-time tracing and logging, and dynamic deployment allows the SOAIF to dramatically reduce the time taken to implement and deliver working processes.

Conclusion

The move toward SOAs will affect all distributed computing in the future. SOAs provide a layer of abstraction over all existing architectures, allowing distributed solutions to be built by composing asynchronous services into composite applications over a network.

Deploying an SOA requires software that provides service-oriented management, integration, security, tools, and processes. While these segments are currently served by individual packages and solutions, they'll evolve into a single SOA and BPM infrastructure solution.

An SOA and BPM framework includes all of the distributed computing functionality an organization needs to develop, deploy, manage, and extend an SOA; these frameworks will come to dominate enterprise computing over the next few years.

About the Author

Deepak Pareek is a creative, forward-looking strategist, coach, author, and speaker capable of translating vision into actions and quantifiable results. He is an expert in enterprise technology with extensive exposure to global management and technology consulting. Deepak has a decade of hands-on experience in multiple technologies and currently works as an advisor and consultant with various top-of-the-summit technology organizations to provide futuristic vision and technology road maps.

deepak.pareek@gmail.com

Pantero 1.3 Shared Data Services

Gain control of your data

■ The moniker "Shared Data Services" has a sexy ring to it, and in a market where service-oriented architecture is the topic du jour, you may be tempted to roll your eyes… but don't. Pantero targets an important segment where business spends $80 billion annually on integration and another $19 billion on manual reconciliation.

WRITTEN BY
PAUL T. MAURER

Shared Data Services?

What are shared data services? A quick tour through the organic growth of Web services at a firm would be a good illustration.

Chances are you have dipped your toe into Web services when your company needed to expose data to the outside world. You may have started with a simple, straight-through service that returned un-enriched data. As you gained experience, you may have enriched the data, significantly expanding the sources of data available, and gone on to enhance the interface to support an industry model. Finally you may have added support for inbound Web service–based transactions that required validation and mapping back into the Corporate Operational Data Stores. This is the world of Shared Data Services.

Pantero

Pantero provides a set of tools for developing and deploying shared data services. The Pantero Designer is an extension to BEA Weblogic Workshop, which allows the user to develop data services. The Pantero Engine executes services, managing logging, and runtime metadata. The Engine runs on both BEA WebLogic Server and IBM WebSphere Application Server.

The Exchange Model

Pantero Data Services operate using a semantic model of the data that Pantero calls the "ExchangeModel." The exchange model reconciles differences between data formats and models among entities. The ExchangeModel contains three model tiers: data sources, data services, and the intervening common model. Pantero defines the common model as "An intermediary model that maps to both data sources and data services and serves as a canonical or reference structure for the data."

You can begin building the exchange model by creating a basic common model (Figure 1). Pantero makes this extremely easy by providing a set of import wizards that can bootstrap a model with an XML Schema, database schema, or UML Model.

Once this is done, you can also import schemas to define data sources and data services.

Next you can enrich the model in a number of ways. You can add simple and computed attributes, define inheritance and containment relationships, and define record keys.

Rules

A powerful feature of Pantero is its support for Rules. Pantero supports four types of rules: Transformation rules, Validation rules, Aggregation rules, and Business rules.

A transformation rule can derive one or more properties of a class from one or more properties of another class.

A validation rule ensures that data meets given constraints, generally defined by schemas.

An aggregation rule resolves duplicate entities or combines separate entities in some domain.

A business process rule evaluates some condition to answer a question posed by a business process.

Combined, the rule types are a powerful and flexible way of insuring the integrity of data flowing through the model.

Data Mapping

Pantero provides the requisite mapping capability. By bringing up the Pantero Project View (Figure 2), the user can select source and target objects to map. This is done by double clicking the arrow pointing to or from the common model, to either the Data Source or the

PANTERO

Company Info
Pantero Corporation

300 Fifth Avenue
Suite 630
Waltham, MA 02451

Phone: +1 (781) 890-2890
Fax: +1 (781) 890-2975
Web: www.pantero.com
E-mail: info@pantero.com

Licensing Information
Per CPU: $40,000

Licensing Information
OS: Windows XP Professional (Service Pack 2)
Hardware: Intel Pentium M Processor (1300MHz) – 1.29 GHz – 1 GB RAM

> Pantero's mapping capability is well thought out and very straightforward to use

Data Service. This action pops up a mapping screen. The screen contains the source objects on the left with the target objects on the right. At this point mapping is a matter of dragging and dropping between available source and targets. Once this is done, a line will appear and indicate a mapping. The line would be dashed if the mapping needs further definition. Clicking on the dashed line would bring up the class view, allowing the user to map attributes between source and target classes.

Pantero's mapping capability is well thought out and very straightforward to use.

Data Service

Where do Web services come into the picture? Well, when you build the Pantero project, you build your data Services as WebLogic Workshop Java controls. This makes the Pantero Data Service available to a WebLogic business process, which can expose the data services as Web services. In Weblogic Workshop this is a matter of a few mouse clicks.

Testing

Pantero provides a Tester component that allows the user to test data services without deploying the application. The Tester establishes a runtime environment in which to execute a data service, insert an XML document into the runtime environment, execute a query, and execute a map. This short-circuits the need to deploy to an application server significantly eases the test/debug cycle.

Impact Analysis

When modifying the Exchange Model, Pantero allows for Impact Analysis. The Impact Analysis component displays lists of objects that might be affected by changes to the selected object. When you select the count for an object type, the Impact Analysis component displays a table that lists objects of that type. This feature definitely comes in handy when your project sponsor wants to know how a proposed set of changes will affect the model.

Documentation

Pantero provides well-written documentation in the form of a Tutorial, User Guide, and of course, online help. The Tutorial is particularly effective because it walks through a fairly involved real-world example using a financial services model, which exercised all of the features. This is not your typical stock quote tutorial. I recently completed a data services project at a large financial institution and this tutorial definitely addresses many of the issues we encountered.

Conclusion

Pantaro's Shared Data Services product is a powerful tool that can give you a level of control and flexibility over your data. I definitely recommend that you check this product out.

About the Author

Paul Maurer is a principal in the Financial Services practice of a leading consulting services company.

paul@paulmaurer.net

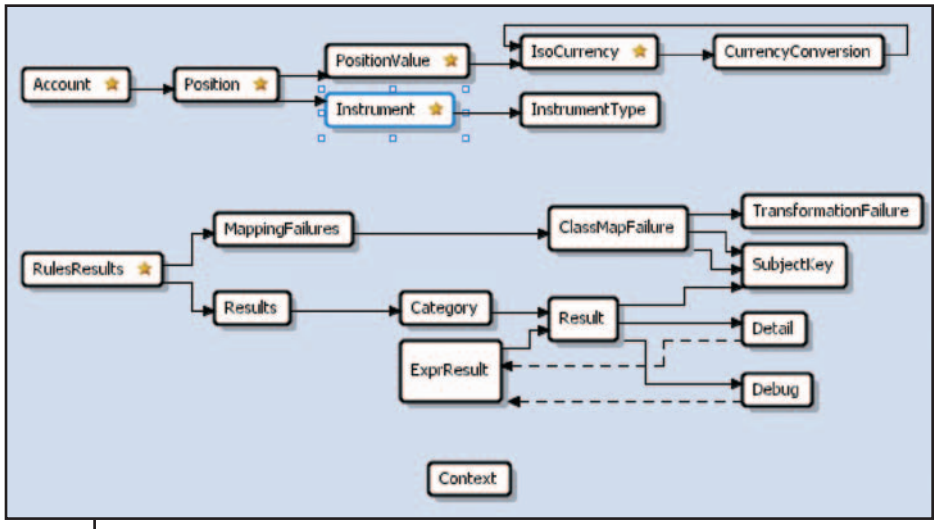


FIGURE 1 **Pantero Common Model**

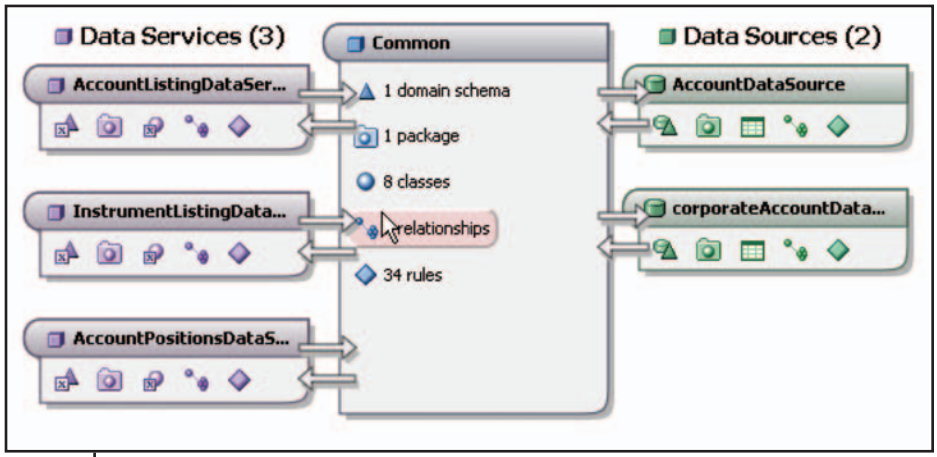


FIGURE 2 **Pantero Project View**

Managing and Integrating Extended Supply Chains

Web services to the rescue

Over the last decade, supply chains have evolved to keep pace with changing business dynamics. This is especially true in high-tech electronics, where companies depend on an extended value chain of component suppliers, outsourced manufacturers, and logistics partners – not to mention B2B-integrated customers – to coordinate vital inter-company design and manufacturing processes. Adding to this complexity, business decisions are more centralized, while outsourced supply networks are increasingly distributed and global.

As a result, extended supply chains face the challenge of being adaptable while integrating with distributed companies in real time to provide advanced decision support. Several new technology developments have been inspired by these challenges; the most notable is managing complex processes using Web services for real-time application-to-application integration.

Advanced Supply Chains Require New Software Architectures

Exchanging information across a distributed network of partners requires a new type of enterprise architecture based on Web services to deliver application-to-application integration. This new architecture relies on a loosely coupled collection of services to achieve the vision of real-time, seamless interaction across multiple companies.

The dynamic nature of this industry, which involves processes and suppliers changing on a regular basis, requires systems to rapidly adapt and respond to changes. The service-based architecture is built on the concept of flexibility, using a system of discovery to search and find the desired services. Service-oriented architectures (SOAs) allow Web-based applications to interact with other Web applications using open standards with little intervention.

While traditional application architectures can address multi-party processes, the custom, tightly coupled nature of the architecture captures the process at a specific moment in time. This approach often requires extensive and time-consuming re-work to address any process changes or additions, limiting its extensibility outside a few key customers or suppliers.

In contrast, a service-based architecture built on defined inter-company processes allows the specifics of the processes to be addressed in a way that acknowledges the dynamic nature of the industry. As such, there is a movement toward adopting services to gain visibility to new data and integrate it into existing inter-company processes. Tie in UDDI to deliver a global directory of services that can handle a particular business signal, provide normalization rules, and new business signals can be used without physically having to re-architect.

WRITTEN BY
DESIKAN MADHAVANUR

WRITTEN BY
ANDRE SRINIVASAN

WRITTEN BY
KARTHIK SRINIVASAN

Inter-company process management spans multiple layers, including partner integration through B2B messages, process models and exception workflows, information aggregation and visibility, and human collaborations through highly specialized portals. The new architecture enables changes to a specific layer without having to go through a long upgrade cycle that impacts other layers in the software stack. For example, to add an exception workflow for managing shipment delays or stock-outs in a vendor-managed inventory (VMI), the process would not require a costly upgrade. Instead, a set of configuration changes to that specific layer keeps the overall process fairly unaffected, thus enabling rapid delivery. The result is a cost-effective model for dealing with changing business processes or large, technologically diverse supply networks. Furthermore, this approach allows companies to leverage their existing back-end ERP, by adding services that can extract and normalize the data.

Developing an SOA to extend the supply chain requires a series of specific services to form the core engine; but additional services can be easily added as needs arise. Typical services that must be connected together include: B2B Gateway Service, Validation Service, Transformation Service, Process Service, Event Service, and Metrics Service. For service providers, all core services must be tied into the Security Service and Monitoring Service.

A simple explanation of the architecture illustrates this concept. From the network edge, the service includes a protocol gateway that can deliver messages to a business process engine where they are expressed, validated, and transformed. However, the same engine that does validation is not ideally suited to model business processes such as demand/supply planning or VMI. These should be expressed as separate services to provide flexibility for change as the business requirements evolve – especially if the business process has been modeled using interoperable descriptions such as BPSS and BPEL. Expressing advanced business processes as services allows for that business process to evolve via configuration (by uploading a new description) or by replac-

XSLmaker

The Smart Way to Transform XML to Web Pages

1. Define XML
2. Drop XML on Web Page
3. XSL and CSS are generated

Done.

100% WYSIWYG

Transform XML => Web pages

Features

Visual tool with Canvas	✔
Auto-generates XML tree based on sample XML data	✔
Drag-n-drop from XML tree directly onto the Canvas	✔
Generates XSL code & generates CSS code	✔

Advanced Features

Merge multiple XML sources	✔
Include live XML/RSS feeds in Web page	✔
Include other Web pages in Web page	✔
Use other Web languages with XSL in Web page	✔
Supply run-time input parameters to XSL Web page	✔

XSLmaker

The only comprehensive XSL development platform for efficient making of professional Web pages and Web applications from XML data. Easy and Intuitive.

FREE trial. Download at www.xslmaker.com

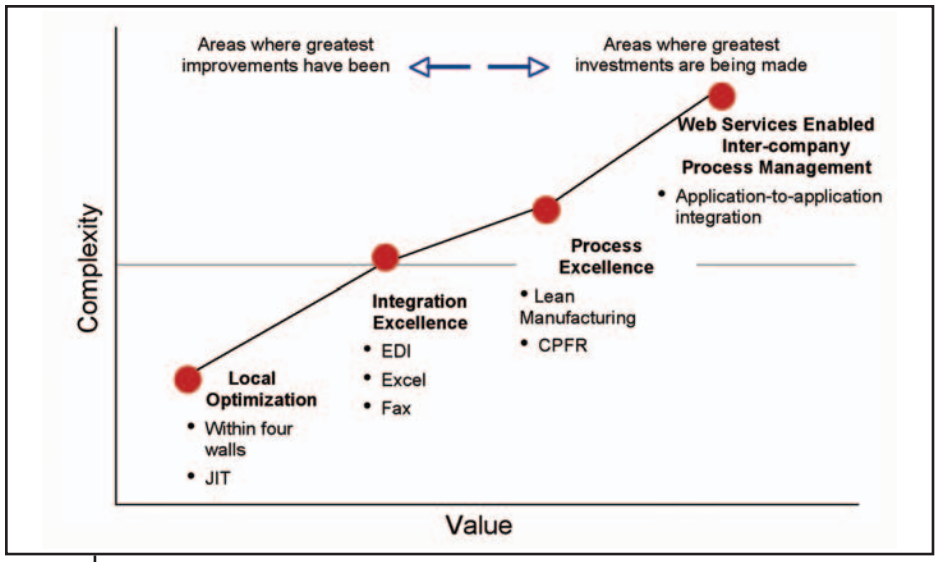


FIGURE 1 **Early supply chains consisted primarily of local optimization restricted to the manufacturing plant or company division. The first technological evolution connected these isolated systems with the external world through both manual and automated mechanisms such as EDI. Next, business processes were added to the data integration, allowing the supply chains to identify and react to the data. Today, Web services help support reaching a vision of real-time, inter-company communication.**

ing the service with one that meets the new requirements.

In addition to accepting (Validation Service), processing (Transformation Service), and delivering B2B messages (Process Service), services are also required to manage, tune, and check the system. The operational infrastructure for this architecture leverages traditional routers, switches, and VLAN fabric, but also incorporates advanced monitoring and security systems to ensure the safety and protection of the confidential process information being exchanged. Furthermore, if a monitoring system is going to correlate information and distinguish between security issues and bugs, it needs to be receiving signals from the security system.

SOA can also extend this infrastructure to cost-effectively manage the on-boarding and testing processes. Since each component is loosely coupled via interfaces, well-defined intercept points exist where test and quality services can be inserted. During the development and on-boarding cycles, these services are used to minimize the amount of human support needed and to allow self-boarding and testing. These services are also leveraged once the system is in production to provide visibility to the quality of the business signal. These signals are picked up by the Metrics Service to provide reports on how well partners are conforming to the business process, thereby allowing improvements to be identified and easily incorporated.

It is important to address the B2B Service, which may be looked upon as redundant. The reality is that business signals are coming from a variety of systems via any number of standards – including RosettaNet, EDI-VAN, EDI-INT, FAX, FTP, e-mail, and custom XML – and require a mechanism to tie them together. The type of signal determines the location of the B2B Gateway Service in the overall architecture – on the edge of the enterprise, on the edge of the network, or as part of the overall core service components. When connecting a legacy system, it may be necessary to co-locate a bridging technology that can extract the data from the system, thus locating the service on the enterprise edge. Once done, the business signal can be normalized for the business process and injected into the overall message flow, thus supporting the vision of inter-company process management.

A robust inter-company process solution or application must be built with an eye toward services to handle the requirements of a complex, multi-party transaction. Services such as those mentioned above are required to add the process logic inherent in a supply chain or manufacturing application to provide capabil-

Evolution of Supply Chain Management

Phase	Complexity	Value
Silo Optimization	**Low** This was primarily done to optimize the functioning of factories or organizations within an enterprise. All integration is limited to in-house legacy systems, thus making integration complexity low.	**Low** Silo Optimization was limited to four walls of the organization, and hence does not account for real-world external constraints from trading partners. Therefore, this system is highly susceptible to sudden changes in customer demand and/or supply.
Integration Excellence	**Medium** Complexity increases as more and more external connections are added. To integrate with companies of varying sizes and technical enablement, supply chains typically support multiple formats for integration such as Excel files, e-mails, EDI, etc.	**Medium** Increase in effectiveness due to standardization of information across the enterprise. This provides an order of magnitude improvement over Silo Optimization. This makes companies more responsive to sudden changes in supply and demand.
Process Excellence	**Medium** The next stage in the evolution incorporates some level of process logic to the information flow. It adds very little complexity over the Integration Excellence phase.	**High** This provides significant value over Integration Excellence. Supply chains now have process controls in place to identify and resolve business problems as they occur.
Web Services–Enabled Inter-enterprise Process Management	**High** High degree of automation with system-system integration and process normalization between enterprises required.	**Very High** Keeps all participants in an extended multi-tier supply chain synchronized. Provides real-time alerts to system changes and converts the supply chain from reactive to proactive.

ity beyond simple B2B integration and e-mailed spreadsheets. Solutions that rely solely on incorporating a few open source libraries while leveraging a file system as an interconnect to accept and report on B2B messages are missing the vision of an SOA. The ability to add new protocols and new process functionality is key to differentiating the more sophisticated Web services architectures. For an application (monolithic or otherwise) to adequately support the complexity of a business process expressed as a service, it must consist of a number of specialized packages, thereby further stressing the need for the software to communicate with a variety of core services.

Leveraging Services for Software Delivery

Along with new software architecture comes another important piece – delivering the software as a service. Proprietary hardware and software have made the current delivery infrastructure rigid and inflexible in contrast to the dynamic and rapidly changing requirements in today's companies. This shift is forcing changes in the underlying computing environment to an on-demand or Software-as-a-Service (SaaS) model.

SaaS is characterized by a combination of software architecture, delivery, and pricing models. This approach leverages SOAs, Web services, and other standards to create a highly modular integration framework. As companies are moving to layered and composite application frameworks to manage their componentized business models, they require a highly flexible and efficient operating environment that removes integration complexity.

There are two types of a SaaS models. The first provides service integration capabilities, which help manage inter-company processes by connecting people, processes, and information; the second provides service delivery and management, which represent a consolidated and logical view of resources across a services network.

An SaaS delivery model allows companies to consolidate their software and to support efforts around a single, integrated, and standards-based configuration, which minimizes the IT resources required for managing a highly complex and dynamic application. Furthermore, SaaS enables companies to rapidly (in weeks rather than in many months or years) configure and implement processes to meet their rapidly changing needs. It also allows companies to address the integration needs of cost-sensitive small- and medium-sized suppliers. Finally, SaaS can provide options for new revenue streams, such as the introduction of complementary software (like add-ons) and services offerings.

Summary

Web services and SOA can now provide companies and their extended supply networks the ability to connect disparate systems through external services – representing the "Holy Grail" of advanced supply chains. Ideally, inter-company process management will allow companies to identify and resolve issues in real time, without human interaction.

While this vision gains mainstream adoption, companies can begin preparing for this shift. A review of the current system architectures may lead to identifying where services can be introduced. For example, it may be possible to leverage a B2B Gateway Service or a Process Service for new business processes. By planning for services as the system evolves, it can slowly evolve to an SOA. Furthermore, a plan should be put in place for:

- Configuration tools, including graphical modelers for business process definition and choreographies
- People and information integration across multiple tiers and various access modes for visibility
- Process transformation for reuse of applications and data
- Business service management
- Security and provisioning
- Resource management

The Service-Oriented Software Architecture

B2B Service or Protocol Gateway
Validation Service
Business Process Service
- Validation
- Transformation Service
- Routing
- Leverage Services

Testing Service
- Message Conformance
- Business Process Conformance
- Simulation

On-boarding
Runtime Management Service
- Replay
- Visibility
- Tracking
- Alerts
- Event Service or Exception Management
- Reports
- Metrics

Solution Management
- Upgrade
- Enhance

About the Authors

Desikan Madhavanur is a principal engineer at E2open and leads product management and design functions for E2open Supply Chain Process Manager and E2open Information Manager, a business intelligence and reporting solution. Additionally he leads a team of solution architects in the analysis and design of complex supply chain solutions for large hi-tech companies. Prior to joining E2open, Desikan was a senior product manager for Supply Chain Collaboration Solutions at i2 Technologies. He has an MS in Systems and Industrial Engineering from the University of Arizona.

desikan.madhavanur@e2open.com

Andre Srinivasan is a principal engineer at E2open and is product manager for the E2open Integration Platform, the messaging and transaction layer for the E2open Software-as-a-Service offering. Prior to joining E2open, Andre helped architect and build the Slam Dunk Network Service, led the design and implementation of CORBA/J2EE Security Service for Borland, and contributed to the implementation of SQL*Net for Oracle. He has a MSCS from the University of Pittsburgh.

andre.srinivasan@e2open.com

Karthik Srinivasan is a director and lead principal engineer at E2open and is responsible for the architecture/design of the E2open solution stack. At E2open, he has contributed heavily to design and evolution of E2open's service-oriented architecture and Web services–based technologies. Prior to joining E2open, Karthik was vice president at Citicorp responsible for development and implementation of solutions for global corporate banking in distributed environments, object-oriented and Internet technologies. He helped create a common platform based on DCE, ENCINA and CORBA for enterprise applications and built mission critical applications for Citicorp global finance. He has a Masters in Computer Science.

karthik.srinivasan@e2open.com

Mindreef® SOAPscope®

Take Control

of your

Web Services

Developers • Testers • Operations • Support

Web services and SOA have changed the rules with the introduction of a new XML abstraction layer.

Though most development organizations have tools and processes for managing code objects, they have little or no help for testing, diagnosing, or supporting the XML portion of their Web services.

Mindreef SOAPscope completes your Web services development toolkit by combining XML-aware tools for developers, testers, support, and operations. This flexible combination gives you the right tool for any diagnostic task, whether working alone or collaborating with other team members.

Start taking control of your Web services by downloading a FREE trial version of Mindreef SOAPscope today!

Mindreef SOAPscope: Named best Web services development tool in the 2005 InfoWorld Technology of the Year awards

Test *No coding required!*
Test service contracts and underlying service code with an easy-to-use, forms-based interface

Diagnose *No more wading through angle brackets!*
Find the cause of a problem with powerful message collection, analysis, and diagnostics tools

Collaborate *No more emailing XML snippets!*
Share test or problem data with others, regardless of their roles or system requirements

Support *No more finger pointing!*
Web services require support at the API level – a real burden for most organizations. Mindreef SOAPscope makes it easy by allowing customers and support staff to share Mindreef package files that contain complete problem data

Download a free version of Mindreef SOAPscope at www.Mindreef.com/tryout

Eight Things SOA Is Not

What not to do in your next SOA rollout

Sometimes when we're faced with addressing a complex engineering problem it's helpful to reflect on antipatterns. Doing so does more than track wrong solutions to common problems; it also focuses the mind on the interaction of the most important elements of the problem domain. This is true for all engineering, not just software engineering. Suspension bridge designers know to be on the lookout for torsional oscillations because of the collapse of the Tacoma Narrows Bridge, but they also better understand the importance of stiffening the structure in general. The goal is to limit the number of times the antipattern emerges and to notice it when it comes around again. SOA uptake is at a point where such a treatment of antipatterns is helpful.

WRITTEN BY
PAUL O'CONNOR

In my job I work with Fortune-500 clients on SOA enablement. Like everyone else who works in this space, I have to try to cut through both the hype and the FUD (fear, uncertainty, doubt) surrounding SOA to right-size solutions for individual business problems. Sometimes the requirements are very narrow, e.g., "We would like to federate with external partners." Sometimes they are very broad, as in, "We would like to re-tool the entire enterprise for more efficient and agile execution." I like to think of SOA as the synergy between WS standards and tools, and an adaptive organizational structure that can leverage the technology to manage complexity better. In fact, when it comes to complexity management, I tell my clients that with an SOA approach we can fashion an architecture to get to their end state that can be wholly contained on one whiteboard.

Here are some fallacies about SOA that keep architects from being able to concisely contain their SOA solutions on that whiteboard.

1. It's Not a Platform

SOA can't be purchased solely from a platform vendor. IBM, BEA, Oracle, SAP, etc. actively market to their customers the premise that more from their product lines = more SOA. This is not to say that these vendors don't have good products that fit into an SOA quite nicely. We are starting to see serious BPEL, WS-ReliableMessaging, and WS-Addressing implementations from platform vendors. However the concept of the platform, which has gained acceptance over the last two years, serves to tie customers into a one-stop shop for both infrastructure and development. This is antithetical to SOA, whose standards make it possible and advisable to pick the best tools available for the task at hand (and no more). Throw in the marketecture avalanche and customers are reluctant to incorporate needed SOA infrastructure elements outside of the platform. By way of example, if you are planning to implement a partner federation model, you can't get there with a platform alone. Worse yet, we are also seeing stuff like BPELJ out of platform vendors, which is a recipe for tie-in masquerading as an unnecessary standard. The moral of the story: take a holistic ap-

proach to SOA infrastructure and avoid platform tie-in. A great place to start is at your favorite open source infrastructure project. Work your way up the food chain from there.

2. It's Not Just a New Name for an Old Development Paradigm

There is a lot of FUD being spread by my fellow Java developers and architects that SOA is just a new services architecture like CORBA or Java remoting. While it is certainly true that loosely coupled RPC service architectures have been around for at least 15 years, SOA is a deceptively powerful simplification of the paradigm. The use of XML and SOAP's realization of an intermediary framework finally gave us true extensibility. Microsoft's signing onto the standards gave us a shot at interoperability – imagine if Microsoft had not signed on and we had a world with XML and SOAP versus MS-XML and MS-SOAP. If it isn't obvious the power that these improvements brought to the table, tell me how to develop a federated security domain of thousands of partners around the world to run a global logistics business, for example, with Java and CORBA. We can do it today with WS security standards and tools, and express the architecture on that one whiteboard.

3. It's Not Custom

Almost all standards are built on other standards and add their own functionality. Standards are the life blood of SOA, even though many are still evolving. The evolving part scares people, but in fact there are very few standards that are truly final. SOA is best served by maniacal adherence to standards. Deviate from them only when they just don't get the job done. As an example, UDDI is probably the most prevalent case in SOA of an often avoided (or at least augmented) standard. However even in this case, registry vendors I am aware of take care to render standard UDDI on demand, and support UDDI as it evolves. So when the time comes to federate registries with other departments or partners via UDDI evolved, you're good to go. If you are eschewing UDDI for a shared Java properties file or other similar service location mechanism, you're on the wrong track. In general, finding a way to support standards in your SOA will keep you healthy in the long run.

4. It's Not Just Hype

Granted, there is more hype around SOA than anything we've ever seen in IT before. A year or two ago a perfect storm of research firms, infrastructure and platform vendors, and integrators emerged under the SOA banner in a post-boom business climate that demanded more functionality from less investment. Throw in the media coverage and the effect was to all but obscure the real technology revolution that was happening in the form of emerging standards and tools.

> "Once you cut through the hype and apply the technology in its pure form to your problem domain, real value is easy to find – and lots of businesses are realizing it now"

It is easy to dismiss the hype as "the next big thing," but real value is being experienced, especially by medium and large enterprises. The value comes in two flavors: the ability to address problems that were previously intractable with the new standards and tools, and the ability to become more agile and better manage enterprise entropy. A good example of the former is the use of SOA security standards to build application security frameworks in a much easier way that support things like digital rights management and portal security. Once you cut through the hype and apply the technology in its pure form to your problem domain, real value is easy to find – and lots of businesses are realizing it now.

5. It's Not a Data Integration Panacea

There is a limit to the amount of data source dissonance that can effectively be managed in an SOA. If you have a bunch of different data sources that have evolved over time into a frayed network, SOA is not going to derive a better integration mechanism for you. It's not going to make it any easier to combine result sets into a workable view, and it might make it much worse if you get the XML translation or granularity wrong. It's bad news to see a bunch of services exposed against different customer data sources, for example, with business processes trying to combine them into a unified customer profile. Look to a virtual data service to generate a holistic view of the data and then call that service from your wider SOA. You can either construct this service yourself or purchase it from one of the emerging EII vendors.

6. SOA Does Not Require More Savvy Developers or Architects

Just the opposite...the whole point of SOA is to manage complexity better and make accommodating new functions easier. This means that developing in an SOA environment is much easier than in a pure-Java or .NET environment. It's all about agility, both in the sense of making it easier to add/change functions, and in getting from development through QA and into production faster. Again, much of the value of SOA lies in putting into infrastructure parts of applications that used to be developed and so the development part necessarily becomes less complex. You will still need a few specialists on the development team, including at least

one SOA architect, and some senior developers that own different component services like the database – they will need to understand how to drive the performance of the service against the database – but the stock SOA developer doesn't need to understand how to do it. This is a big change and cost savings. Most good J2EE developers are intimately familiar with driving performance of their code against a database. It's among the first things that they learn. SOA obviates the need for all developers to know much more than how to leverage your firm's SOA tools against your services. I've even seen clients successfully retrain mainframe programmers for SOA process development that have no foundational OO language skills whatsoever. So, if your HR department is putting out requirements to job boards that include your old platform spiel *plus* your new SOA stuff, you're on the wrong track.

> "I like to think of SOA as the synergy between WS standards and tools, as well as an adaptive organizational structure that can leverage the technology to manage complexity better"

7. The Biggest Challenge in Moving to an SOA Is Not Technical

It's organizational. Realigning political boundaries and responsibilities and establishing a governance regime proves difficult for most development groups. If your group is organized around a set of siloed applications then you can envision the problem. There has been a lot of debate recently about how best to approach this. Should it be addressed top-down (CIO puts forth an SOA vision and tries to convince the business to pay for the retooling) or bottom-up (developers and architects interested in making things better create an SOA groundswell)?

In my experience it's both and neither. It's a runaway process of continuous improvement where the development team identifies some low-hanging fruit to attack with SOA, which provides good value to the business, thereby leading to an ever-lengthening leash to build more SOA aspects. The alternative to this approach almost never works: a meeting with the business in which you have to ask for the big investment in SOA tooling and training to be able to deliver what they think you already should be. If you get this far, the remaining trick is to generate an infrastructure that doesn't look piecemeal. Look to open-source (or already paid-for) tools and also consider proposing to address a problem of great business need with tools that can be reused for other SOA aspects.

8. It's Not ESB

ESB evolved as a combination of hype ("We're not a proprietary integration broker, we're the bridge to SOA…) and necessity ("If SOA mediation standards aren't there, how do I do real integration?"). Not to rehash the ESB/SOA debate here, but I lose more gusto for ESB every week as support for SOA mediation standards are announced. If I can do with SOA standards what ESB accomplishes but with fewer of them, I go with the pure SOA every time in the hope that as my enterprise evolves I can repurpose and realign at a much finer granularity. I bristle at the thought of waiting for the next release of someone's integration platform to get a rendering of a new standard. At the same time, I understand the need and presence of ESB in the real world. We just need to understand that it's not SOA.

About the Author

Paul O'Connor is the SOA principal architect for Anexinet (a leading NE boutique integrator) and is currently doing major Web services and SOA architecture for Fortune 100 clients in New York City and Philadelphia. Previously he was chief architect for Damascus Road Systems, specializing in security architecture. Paul is a JBoss Certified Consultant and a contributor to several open source projects.

poconnor@anexinet.com

WebRenderer™

Standards compliant embeddable web browser component

WebRenderer is a cutting edge embeddable Java™ web content rendering component that provides Java applications and applets with a fast, standards compliant HTML and multimedia rendering engine. WebRenderer provides a feature-packed API including complete browser control, a full array of events, JavaScript interface, DOM access, document history and more.

Why WebRenderer?

- Standards Support (HTML 4.01, CSS 1 & 2, SSL, JavaScript, XSL, XSLT etc.)
- Exceptionally Fast Rendering
- Predictable Rendering
- Scalability (deploy in Applications or Applets)
- Security (based on industry standard components)
- Stability and Robustness

Embed WebRenderer to provide your Java‘ application with standards compliant web content rendering support.

To download a 30 day trial of WebRenderer visit
www.webrenderer.com

JadeLiquid™ Software

StrikeIron OnDemand Web Services for Microsoft Excel

Web services for the masses

There was a time when the term "spreadsheet" simply meant columns of data that were summed up. Although this is all the functionality some people require, Excel has evolved into a much more powerful tool. Users now have the ability to create complex formulas with built-in functions such as MAX, SIN, and SUMIF. They can link workbooks together and not only import data directly from a database, but mirror many database capabilities using VLOOKUP and Pivot Tables. Business administrators can use Excel to automate many of their daily tasks, where they used to require the IT department to develop a database and application. With this additional power, concepts such as primary and foreign keys are no longer foreign to the common worker.

WRITTEN BY
KEITH FROHNHOEFER

Perhaps the reason Web services has not spread like wildfire (as was first expected) is because it has remained in the IT arena. Up until now a person had required a developer to create an application for him that leverages the power of Web services. Even the idea of what Web services are and what can be accomplished with them remains a mystery to the masses. For instance, a travel agent may wish there were an electronic way to reserve hotel rooms with any chain, but he probably doesn't realize that Web services exist and may, one day, be the answer to his prayers. How will non-programmers become familiar with the concept of a standard way of interoperating between different software applications running on multiple platforms?

StrikeIron's product, On-Demand Web Services, may be not only a useful tool, but an introduction to Web services for many nontechnical people as well. It grants the use of Web services to Excel users without any programming required. It doesn't take long to get up to speed. After installation, a "StrikeIron" menu will appear in the Excel menu bar. This will provide you with access to a Web service manager dialog. This dialog can be used to search for available Web services or to locate a specific one. Once you select a Web service you are provided with graphical trees representing the Web service input and output. Now it is just a matter of dragging the input parameters and output fields to your spreadsheet. Input parameters could represent a Web service license key or stock ticker symbol. Output fields could represent the current stock price. These Web service cells are automatically formatted with a comment describing the value that will reside there and what Web service it originates from. You can rename labels and reformat cells. Then simply refresh the Web service from the StrikeIron menu. That's it – no programming required!

The product is even smart enough to fill in multiple rows for one output field, if the Web service provides multiple values for it. So if your spreadsheet requests information on a particular recipe, you simply have to drop the Ingredients field onto your spreadsheet. When you refresh the Cookbook Web service, each ingredient

STRIKEIRON

Company Info
StrikeIron, Inc.

2520 Meridian Pkwy
Suite 500
Durham, NC 27713 USA

T: +1.919.405.7010
F: +1.919.405.7025
email: sales@strikeiron.com

Web: www.strikeiron.com

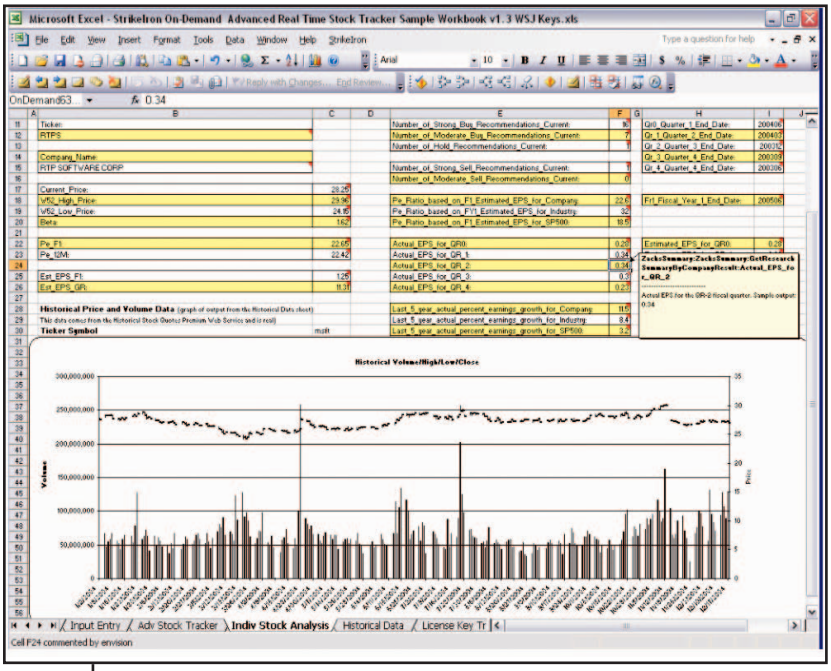


FIGURE 1 **The upper portion of the spreadsheet accesses a Web service for displaying current stock information. The graph at the bottom accesses another Web service that provides historical stock information.**

will reside in a row under the Ingredients column. StrikeIron makes it so simple and intuitive that you wonder why this functionality isn't included in the Excel application.

Individual Web services used in the spreadsheet can be customized in multiple ways, such as when to refresh (manually, on start up, on calculation, or every X minutes). Using and refreshing Web services is something to consider, especially when you are paying for it. Any Web service can be used with the product, but StrikeIron also provides a long list of subscription Web services that are paid for on a per-hit basis. The list includes Stock Quotes, D&B Business Information, Do Not Call Verification, Address Verification, and Tax Rates. Pricing is set for individual services. For instance, Real-Time Stock Quotes Basic starts at 1000 hits per month for $9.95 and ends at 1,000,000 hits per month for $359.95. It should be noted that some services require multiple hits per call to retrieve information. This service uses five hits per successful result.

You can incorporate multiple Web services within one spreadsheet. I should mention that the sample workbooks provided by StrikeIron do a good job of showing how Web services can be used in a spreadsheet and complement each other. For instance, one workbook uses the RealTimeStockQuotes and HistoricalStockQuotes Web services to display current detailed stock information alongside graphs representing historical stock information for a company.

A 30-day trial of the software is available on the web (www.StrikeIron.com). An annual subscription to the software is $99.95. A perpetual license is $299.95. The software requires Excel 2002 and the .NET Framework with SP1.

About the Author

Keith is lead application developer at Envision Consulting Group. Envision provides consulting services and software solutions to the pharmaceutical industry. Keith enjoys programming in Java.

KeithFrohn@hotmail.com

Managing Enterprise Data Complexity Using Web Services: Part 1

Data services architecture

Business data is one of the most critical components of the IT portfolio of any enterprise. Most e-business applications are responsible for reading and writing business data in some form or other. Therefore, the efficient storage, retrieval, and management of the data constitute a challenging problem in all organizations.

WRITTEN BY
DR. SRIRAM ANAND

Enterprises with multiple lines of business spread over multiple geographies have critical data stored in multiple, scattered databases. In many cases, the scatter of core data is proportional to the size of the enterprise IT portfolio. Organizational growth, such as mergers and acquisitions, compound this problem. Such companies may have heterogeneous data environments with varied schemas and they may contain redundant data elements. This data may be static reference data, such as personal customer information or geographical data, common business data, or common external data such as market data. This can lead to serious inefficiencies and consequently higher costs because of the overhead in accessing/updating data in multiple databases using different mechanisms. These issues lead to an incomplete view of core business data such as customer information that can cause inconsistent user experience and the introduction of risk. Apart from this, interoperability between business lines is often difficult and error-prone. In this paper, the overall case for shared data services is developed with a reference implementation based on financial services. In a future work, this approach will be applied to a different industry vertical and subsequently, specific implementation concerns will be addressed along with migration strategies.

Business Problem

Consider an example from financial services where some of the larger players have multiple lines of business, such as personal banking, personal insurance, and credit cards apart from scattered geographical presence. Common data, such as customer personal data, may be maintained in multiple repositories and checked for consistency in separate applications using proprietary logic. This can lead to issues in synchronization, user experience, etc. A typical problem that can occur is related to a portal through which a customer may update his or her personal information through one LOB, such as personal banking. Due to synchronization lag, this data may not percolate to the database associated with other business applications, so the customer may see outdated information in other LOB applications such as personal insurance. The key issue is that there is no holistic view of core customer data. This can lead to issues in satisfying business requirements. Because of this, other applications such as CRM, risk management, etc., do not have a comprehensive and current view of core business data. Figure 1 illustrates a detailed business context where this situation is applicable.

Note that this represents a typical financial services organization with offerings in the insurance, asset management, and brokerage domains. In addition to this, the company has also expanded geographically to provide insurance offerings in Europe. A multitude of applications access and update critical data to multiple databases in each separate line of business and geography. Dynamic business needs and the push from competition have created drivers for cross-business integration and the desire to provide users with an integrated experience. This in turn has led

to increased demands for the availability of core data from other lines of business and the need for the data to be in sync. Currently, this is performed by using custom synchronization routines that are unstable and difficult to maintain. Apart from this, these types of solutions are not scalable given the expanding nature of business, the nature of different business units, and the technology platform that may be in use in multiple lines of businesses. In addition to the Web applications, note that there is a team of dedicated back-office representatives who interact directly with customers to resolve various issues, set up, etc., and directly update databases through a suite of desktop applications. An IVR channel also exists through which customers may directly make changes to core personal data. These update channels present an additional challenge to the integration of data and the creation of consistent views of core corporate data such as customer reference data.

To summarize the key challenges of this problem domain:

- Critical business data is located in multiple repositories
- Data has multiple channels of update, including "back door" updates
- Data needs to be synchronized between repositories
- Needs and usage patterns of data are diverse across systems
- Different lines of business have different technologies, hence it is difficult to create a common data dissemination strategy
- Performance requirements of data access are increasing due to the evolving nature of business
- In the long term, it is more beneficial to move away from siloed data models from the cost and agility perspective

Based on this problem statement, let's examine some options for tackling these problems.

Solution Tracks

Based on an analysis of the problem statement above, one can propose two parallel tracks of activities that can mitigate the problems mentioned above over a period of time. Let's examine the tracks below and later we'll look at a specific track in additional detail.

Develop a comprehensive DSA (Data Services Architecture)

In order to develop a DSA, we need to focus on the following areas, keeping in mind the specific criteria that need to be addressed in each area.

- Integration and consolidation of data
  - Work with the technology and business groups to develop unified schema for common data
  - Identify target databases for storage of common data
  - Develop a coherent migration strategy to attain the new DSA
- Dissemination of data
  - Needs and usage patterns of data are diverse across heterogeneous consumers
  - Develop a scalable strategy that can accommodate new applications with minimal turnaround time
  - Ensure that data consumer applications are insulated from issues with data management

Rationalization of processes

- Data has multiple channels of update, including "back door" updates. This needs to be examined and modified to make sure that updates are performed consistently.
- Develop a strategy that will move away from siloed data models in the long term.

Data Services Architecture

In this article, I will focus on the data services architecture. The other aspects mentioned above are very important and can fill up journals on their own. Specifically, issues such as the development of a common data model can be very challenging from the organizational perspective, and much more so than from the technical standpoint. I will focus instead on the architecture that can provide a viable option for integration and dissemination of the data given the discussion in the solution tracks section discussed above.

The crux of the solution option in this regard involves the development of shared data services. The principle behind shared data

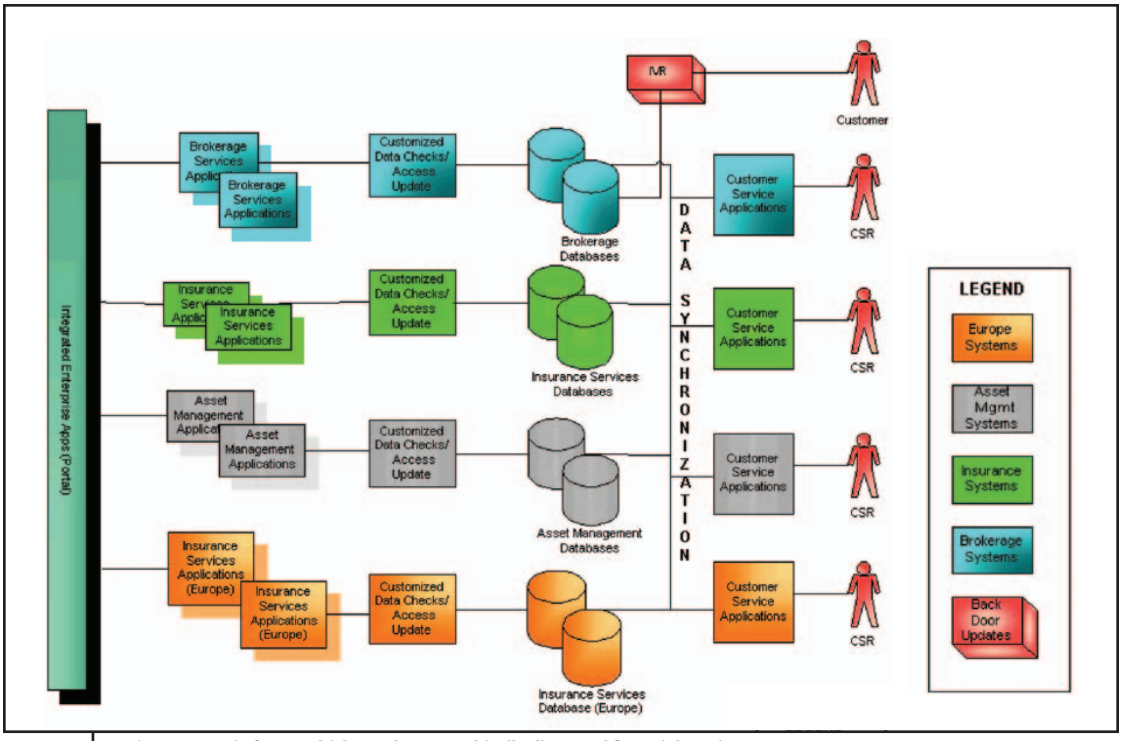


FIGURE 1 | **Business scenario for a multichannel, geographically dispersed financial services company**

services is the consolidation of common data along with the development of interfaces for data dissemination using open standards. Web services provide a viable option in this regard due to the ubiquitous nature of the Internet, and more importantly, the lack of constraints on the consumer end from a technology standpoint. Developing the shared data services will help alleviate some of the problem areas by:

- Eliminating redundant data by gradual reduction in the scatter of core data
- Standardizing mechanisms for data access/update
- Standardizing formats for common data across the enterprise
- Modifying business applications to use the new services for all data

This will significantly reduce the costs incurred for data access in multiple lines of business. Specific activities that reach toward the adoption of shared data services are as follows.

- Develop shared data services that can retrieve information for a set of related applications. Each client application uses the data service, which manages the relationships between the databases and related systems. The services will become the only channel for read/update of all data for all business applications.
- Rationalize schemas for common data across databases.
- Design service contracts based on the needs of individual LOBs/client applications.
- Provide information on demand (in response to service requests) by optimizing performance and caching heavily accessed data.

The specific features of the proposed DSA are as follows.

- A centralized database that maintains data common to multiple lines of business. This database will need to be created as discussed above, by analyzing schemas and formats of data required by current and future business needs. This database will become the system of record and the owner for the common data.

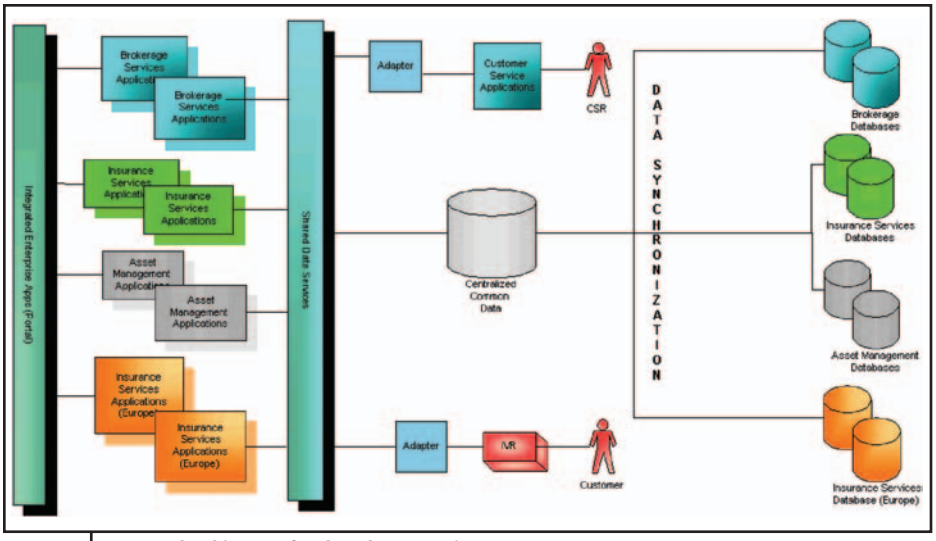


FIGURE 2 **Proposed architecture for Shared Data Services**

- A set of Web services that manage data access/update for all databases. These services will manage all data access/update and will become the de facto data access layer for all applications. The services will provide access to multiple data sources; it will be the responsibility of the consumer business applications to manage heterogeneous transactions appropriately based on the associated business process.
- The Web services will provide data at a granularity that is dictated by the business use cases. The Web services should not be designed based on existing queries or views. This is a very important success factor for the successful adoption of the data services. Domain decomposition should be performed to determine the use cases and their specific data needs. One caveat in this regard is that most applications will require some modification to enable them to use the new data services. The specific mechanism to perform this must be a part of the detailed migration strategy that is developed for the adoption of the data services.
- The IVR and CSR applications will interface with dedicated adapter applications that will communicate with the data services. This will ensure that the services become the only gateway to critical data. The service implementations will ensure that data is consistent at all times between the repositories.
- Line of business data continues to reside in an existing database. Depending on the specific line of business and the problems (or lack thereof) associated with the business data, this data will continue to remain in the existing databases. Physically the data may be migrated to one common platform if it makes sense from a strategic vendor management or licensing perspective. However, the important point to note is that data will be segregated on the basis of usage by lines of business with the shared data services layer simply providing a uniform mechanism of accessing the data.
- Note that there is an element of data synchronization present in the current architecture as well. This is due to certain elements that may be related to specific data items required by legacy applications in different lines of business. It may be more difficult for these applications to invoke a Web service for a specific attribute, than it would be to pick up the data through some synchronization routines. However, it needs to be pointed out that the synchronization routines developed here will be a part of the DSA and therefore should be implemented in a consistent manner along with the Web services.

Why Web Services?

It is important to point out the rationale for using Web services as the backbone for the data services architecture. There are other options for implementing this DSA such as an ETL tool, an EAI tool, or custom integration. The reasons why these are not suitable for this case are related to the factors mentioned in the Solution Tracks section above. These options would be viable for data integration, although in the case of the ETL tools or the EAI tools, there is the potential for vendor lock-in. Apart from this, it is critical

to examine the issue of the heterogeneous consumers of data. It is imperative to provide data through an open channel that does not impose any constraints on the collaborating applications due to technology or client libraries. This is applicable to existing applications and also to future consumers about whom we have limited knowledge at the present time. Web services represent a viable choice to satisfy these requirements given their open standards and the ability of any technology platform – .NET-based, J2EE-based, or Mainframe – to consume them.

Potential Roadblocks

The path to Web services nirvana is not easy and the organization needs to make a firm commitment to this cause and be aware of certain critical factors described below.

- The DSA must be implemented in a top-down manner by focusing on the business processes and the role of data in the process fulfillment. A heavy investment must be performed in upfront analysis to understand business processes and data usage.
- Service designs should not be implemented without complete support for the interfaces from all stakeholders. If not followed, this will later lead to an undesirable proliferation of services.
- An acceptable governance process must be implemented with clear ownership of services. The best approach is to create a dedicated data governance council that facilitates the entire process and works with the business as well as with technology groups on a continuous basis.
- Service granularity must be designed based on the "get only what you need" principle. In the interests of performance, there is limited value to be gained in designing "one size fits all" type services. As mentioned earlier, the interfaces must be designed keeping in mind business processes associated with various business units. For example, service interfaces must have flexible data paging capability in the interfaces. This will ensure that consumers can control the amount of data returned to them and the workload of the services can be optimized.

Conclusion

In this article the case for shared data services has been analyzed. Shared data services can significantly increase reuse and developer productivity while providing consistent performance and highly available data across business units. There are technical and organizational challenges with respect to the migration to shared data services. A prudent approach to migration would involve migration in stages, with new applications moving over to use the new services before decoupling older applications from existing data access methods. Readers can use this article to come up with strategies to centralize data management and move from piecemeal, "band-aid" type solutions. Apart from this, they can also develop a data architecture that can reduce the complexity of managing data residing in multiple data sources and become flexible toward the needs of heterogeneous data consumers. Finally, we can conclude with the thought that implementation and rollout of a comprehensive data services infrastructure will require significant upfront investment in time and cost; however, the short- and long-term benefits should justify the cost. In a future article I will demonstrate the applicability of Web services in the development of enterprise-level dashboards for management reporting.

References

- *How Do You SOA Enable Your Data Assets?* Jim Green, DM Direct Newsletter, October 15, 2004
- *12 Steps to Implementing a Service-Oriented Architecture*, David S. Linthicum, White paper – Grand Central Communications, October 2004
- *Data Services for Next-Generation SOAs*, Christopher Keene, *Web Services Journal*, December 2004
- *Architecting Data Assets*, Jeff Schulman, Gartner Research, August 2004
- *XML, Web Services and the Data Revolution*, Frank P. Coyle, Addison-Wesley, 2002
- *An alternative architecture for financial data integration*, Alberto Pan et al, Communications of the ACM, 2004

About the Author

Dr. Sriram Anand has over 14 years of work experience in industry and research and holds a Bachelor's degree from IIT-Madras and a PhD from SUNY-Buffalo, USA. Sriram is a principal researcher in Software Engineering and Technologies at Infosys Technologies, Bangalore. His career in IT has spanned various roles including component developer, technical lead, senior architect, and development manager. Sriram is experienced in designing enterprise architectural strategy for leading U.S. companies in the financial services, retail, and pharmaceutical domains. He has also published several articles in leading journals and participated in conferences in the engineering and IT domains.

sruram_anand@infosys.com

*Offer subject to change without notice

Learn Web Services. Get a New Job!

Only $69.99

1 year (12 issues)*

* Newsstand price $83.88 for 1 year

Get Up to Speed
with the Fourth Wave in Software Development

Subscribe

ONLINE
www.wsj2.com or

CALL
888 303-5252

*Offer subject to change without notice

The Best .NET Coverage Guaranteed!

Subscribe today to the world's leading Web Services resource

- **Real-World Web Services**: XML's Killer App!
- How to Use **SOAP** in the Enterprise
- Demystifying **ebXML** for success
- **Authentication**, **Authorization**, and **Auditing**
- **BPM** - Business Process Management
- Latest Information on **Evolving Standards**
- Vital technology **insights** from the nation's leading Technologists
- Industry **Case Studies** and **Success Stories**
- Making the Most of **.NET**
- **Web Services Security**
- How to Develop and Market Your Web Services
- **EAI** and Application Integration Tips
- **The Marketplace**: Tools, Engines, and Servers
- Integrating **XML** in a Web Services Environment
- **Wireless**: Enable Your **WAP** Projects and Build **Wireless Applications** with Web Services!
- Real-World **UDDI**
- **Swing**-Compliant Web Services
- *and much, much more!*

SYS-CON MEDIA

Web Services JOURNAL .NET J2EE XML

SYS-CON Media, the world's leading *i*-technology publisher of developer magazines and journals, brings you the most comprehensive coverage of Web services.

Web Services–Oriented Architecture

PRESENTED BY
WSJ EDITORIAL BOARD

EDITTED BY
AJIT SAGAR

Starting this month, to meet the needs of our readers we are introducing a new page for Frequently Asked Questions that will be addressed by our esteemed editorial board. No topic is too basic or too advanced to be discussed here. Please send us your queries and we will provide you with accurate and up-to-date information on topics related to XML, service-oriented architecture, and Web services. If you have a question pertinent to this space, please send it to WSJFAQ@sys-con.com.

1. What is the relationship between service-oriented architecture and Web services? Doesn't SOA automatically imply Web services?

A common misconception is that SOA implies Web services. This is grossly incorrect. SOA is a concept. Web services are a technology or technology platform. Service-oriented architecture is a methodology for achieving application interoperability and reuse of IT assets to enable business agility. It encapsulates governance, process, modeling, and tools to align business functionality requirements with the technical capabilities of the underlying software platform. The basic idea behind SOA is the realignment of application architecture around reuse, efficiency, technology independence (decoupling of technology and business functionality), and flexibility to enable maintainability. SOA is neither a technology nor a technology standard; rather it's a high-level concept that enables the development of a service-based architecture blueprint.

SOA, as the name implies, is based around the concept of a service. The building blocks for achieving an SOA are an application front end, a service that is exposed via a service bus, and a service repository. In today's industry, there are four trends that have converged to facilitate a service-oriented enterprise – Business Process Management (realization of the business), SOA (enablement of business agility), Web services (a natural platform to service-enable the enterprise), and XML (the underlying technology). This is shown in Figure 1.

While SOA is a concept, in order to implement an SOA, a technology platform is required. The most feasible option for this in the industry today is Web services. The requirements for realizing an SOA are the standards, a standard interoperability protocol, a standard API for defining the services, and a universal repository. Web services constitute a technology platform that fulfils all of these requirements.

2. What is a Web Services–Oriented Architecture?

When SOA is implemented using the Web services technology, it is called a Web Services–Oriented Architecture (see Figure 2). In his WSJ article "Web Services–Oriented Architecture: A Critical Technology" (http://webservices.sys-con.com.read/39471.htm), Ashish Deshpande describes the challenges in building a WSOA. While Web services are not the only option for building SOAs, they are the most prevalent and feasible option available to organizations today. Gartner predicts that by 2008, SOA and Web services will be implemented together in more than 75 percent of new SOA or Web services projects.

3. Can SOA be implemented without Web services?

Absolutely! SOA is not a new concept – it's just that technology has recently come full circle to provide optimal platforms for building service-oriented architectures. In fact, the main message behind SOA is not the "Web" but rather the "service." The main objective of SOA is to help organizations move towards a service-oriented enterprise (SOE). An alternative to leveraging Web services is to base your SOA around messaging products. In their current incarnation, messaging vendors are offering Enterprise Service Buses (ESBs) that enable accessing services in an SOA via different means: API calls, messaging, as well as Web services.

About the Author

The WSJ Editorial Board comprises distinguished professionals in the technology field. Here they share their expertise with the readers by answering frequently asked questions about Web services–related topics.

WSJFAQ@sys-con.com

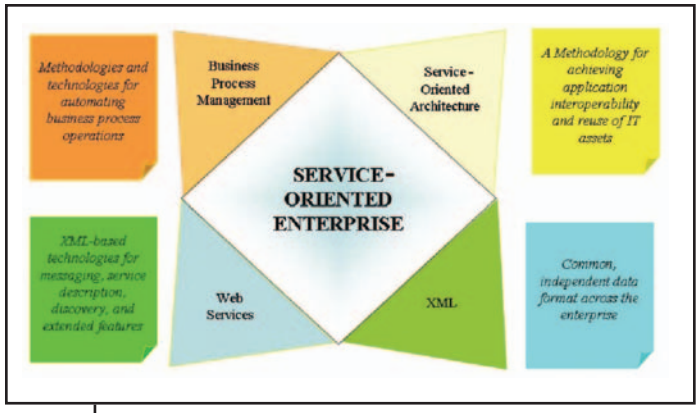


FIGURE 1 Trends converging to a service-oriented enterprise

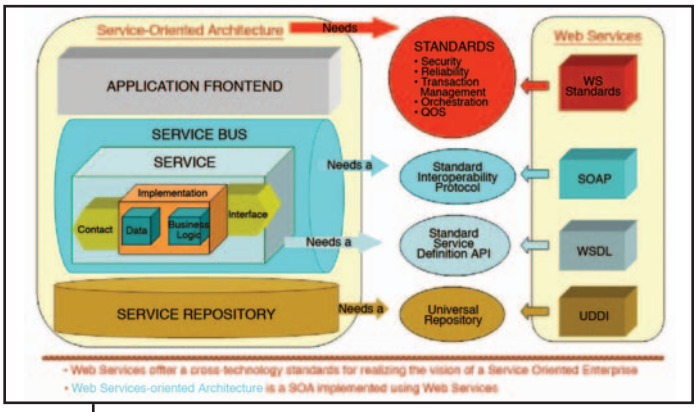


FIGURE 2 How Web services can be used to realize an SOA

Subscribe Today!

– INCLUDES –
FREE DIGITAL EDITION!
(WITH PAID SUBSCRIPTION)
GET YOUR ACCESS CODE INSTANTLY!

The major infosecurity issues of the day... ***identity theft, cyber-terrorism, encryption, perimeter defense, and more*** *come to the forefront in* **ISSJ** *the storage and security magazine targeted at IT professionals, managers, and decision makers*

Editorial Mission

Greater Collaboration and Efficiency Through Education

✓ ***ISSJ*****'s editorial mission is to showcase proven solutions** that will guide, motivate, and inspire senior IT and business management leaders in the planning, development, deployment, and management of successful enterprise-wide security and storage solutions.

✓ ***ISSJ*** brings together all key elements of data storage and protection, and presents compelling insight into the benefits, efficiencies, and effectiveness gained by focusing on these two critical areas of IT simultaneously.

✓ ***ISSJ*** is an objective, critical source of information that helps storage and security managers make informed management decisions about what is working today, and what they need to plan for tomorrow, and is the only publication that focuses exclusively on the needs of IT professionals who are driving the enterprise storage architecture/infrastructure while pursuing and incorporating the latest security technologies.

✓ ***ISSJ*** achieves our mission by delivering in-depth features, practical "how-to" solutions, authoritative commentary, hard-hitting product reviews, comprehensive real-world case studies, and successful business models, written by and for professional IT storage and security practitioners.

SAVE 50% OFF!

(REGULAR NEWSSTAND PRICE)

Only $39.99 ONE YEAR 12 ISSUES

www.ISSJournal.com

or

1-888-303-5282

SYS-CON MEDIA

The World's Leading i-Technology Publisher

Designing Services for Performance – Part I

How to make sure your services keep up

Performance is often an afterthought when building new systems, and I'm finding that services are no exception. Truth be told, most services out there just function. They are not optimized to scale, and SOAs are running into walls as those services hit the upper limit. If we don't learn to solve this problem, many SOAs simply won't get off the ground.

There is indeed a right way and a wrong way to design a service. Also, there are things beyond your control that you must consider during your design. As with anything else, you need to do your homework, allow enough time for design, and do some experimenting and proof-of-concept testing to determine the best path.

WRITTEN BY
DAVID LINTHICUM

First, Know Your Service Patterns

A few patterns are beginning to emerge. We can categorize them into larger buckets such as legacy abstraction, simple composites, complex composites, and new autonomous services.

Furthermore, we can put them into behavioral subcategories such as transactional, data services, and lightweight and heavyweight services. Notice there is no mention of fine grained and course grained; we'll get to that next month.

Legacy abstraction services are services built on top of existing services, including elderly technology such as Cobol and CISC on the mainframes, or perhaps services liberated from mini computers, or even enterprise class Unix systems. You can toss ERP and CRM applications into this mix as well.

The notion is that you somehow are able to externalize these internal processes as services and leverage them as modern Web services, no matter how ugly and arcane the interfaces are.

Simple composites are one or two services that are bound together in a new service. *Complex composites* are many layers of services that are bound together, perhaps a composite that's made up of other composites. *New autonomous services* are services that are created for a single purpose such as a Web service, and are typically not based on other services (non-composite).

Transactional services can be a simple or complex composite, or even new autonomous, but they support transactional characteristics including ACID. For those of you who have not seen ACID as many times as I have, Atomicity refers to the "all or nothing" quality of transactions. The transaction either completes, or it does not. Consistency refers to the fact that the system is always in a consistent state, regardless of whether or not it completes the transaction. Isolation refers to the transaction's ability to work independently of other transactions that may be running in the same environment. Durability means that the transaction, once committed and complete, can survive system failures. With new standards such as WS Transaction, the way in which you build a transactional service should be more consistent. For now, developers are taking their own unique approaches, typically leveraging TP monitors or application servers.

Data services, as you might expect, are services that are built to produce and consume data. These could be Web service abstractions on top of call level interfaces, or simple services exposed out of an ERP system that produces data. These are very simplistic services, with schemas, access controls, and the encapsulated data. These services are almost always built on top of a relational database, but other database types are leveraged as well. Moreover, through a data services abstraction layer, you can emulate database types to meet the needs of your SOA.

Lightweight services, as the name implies,

> "As with anything else, you need to do your homework, allow enough time for design, and do some experimenting and proof-of-concept testing to determine the best path"

means that you're doing things with a light volume (typically fewer than 10 invocations or messages-per-second), and the size of the message that the service is passing is small (typically less than 50 KB). *Heavyweight services*, in contrast, do heavy volumes (greater than 10 invocations or messages-per-second, but more typically 100-300 invocations and message-per-second), and can transmit and consume huge messages.

Second, Experiment and Test

Many of those who focus on the discipline of performance within complex distributed systems such as SOA will first steer you toward modeling. Unfortunately, we don't know enough about how services behave to model how they will perform, so it's a good idea to test the services that will make up your SOA before you build your performance model; otherwise, you're just guessing.

So, how do you test services you've not yet built? It's called a "proof-of-concept," meaning you stand up very raw and simplistic versions of the services (either existing abstractions or new services) for the purpose of proving that they work and to illustrate their operational characteristics. This is typically done in parallel with existing design work, and the proof-of-concept is largely a throw away after you gather your data, but nonetheless it's important to your understanding of the final product before you complete the design and development.

Testing services, even proof-of-concept services, means that you simulate operational characteristics during the test, or, how you intend to leverage the service. You do this by building or buying test harnesses that can load the service as needed for testing. You should utilize low use, medium use, and high use scenarios to determine how the service behaves under an increasing load, and make sure you have some sort of monitoring mechanism to gather the data for analysis.

What you'll find, in most cases, is that the service will reach a saturation point where performance drops off significantly as the load increases. The saturation point is largely dependent on the patterns of the service. For instance, transactional service should be able to support a much higher load than lightweight services.

Next month we'll expand our discussion to the next steps, including: creation of a performance model, and design, monitoring, and optimizing performance of services after they exist in your SOA.

About the Author

David Linthicum is the CTO at Grand Central Communications (www.grandcentral.com), and a leading expert in the application integration and open standards areas. He has held key technology management roles with a number of organizations, including CTO of both Mercator and SAGA software. David has authored or coauthored 10 books, including the groundbreaking and best-selling *Enterprise Application Integration* released in 1998. His latest book is *Next Generation Application Integration, From Simple Information to Web Services.*

dlinthicum@grandcentral.com

THE FAST LANE TO INTELLIGENT INTEGRATION

Are you being asked to deliver more fixed price application and data integration projects? Are those integration projects taking too long and increasing your risk of not delivering on budget? Do you find that the cost and complexity of the integration products available in the market prohibitive? It doesn't have to be that way...

With the FusionWare Integration Server, Solution Providers like you are:

- Expanding the reach of their business with an affordable and easy-to-use Web services solution built entirely on open standards
- Enhancing their existing customer loyalty by providing integration technology which helps them leverage their existing systems quickly and affordably
- Minimizing complexity and risk of integration with a product that runs on any of the major operating systems and requires no additional infrastructure components
- Increasing their sales and profitability

Find out how! Contact FusionWare today at: **sales@fusionware.net.**

FusionWare Corporation

www.fusionware.net

FusionWare:
Cost of Ownership in Integration: A New Perspective

What is *cost of ownership* as it relates to integration? One thing is certain: the cost of integrating disparate stovepipes of data and applications has typically been prohibitive, due to months of integration and development work. But it doesn't have to be that way…

Most of the costs around data integration projects come from far more than software licenses and hardware. Many organizations have found that the leading integration toolsets require armies of IT staff and resources to configure and build. And don't forget the long-term costs such as maintenance. If it's necessary to hire a team of experts to build the application, these individuals will be required on a long-term basis to maintain the application.

A recent survey of 1,000 organizations conducted by Webservices.Org finds integration to be a primary driver – and benefit – of information standardization efforts. However, integration is also the biggest headache for most companies due to the heterogeneity of most IT infrastructures.

How are organizations addressing the need for more seamless integration, without washing away any cost savings on the integration effort? **FusionWare Corporation** (www.fusionware.net), an emerging leader in data and application integration, enables companies with complex application and data integration requirements to quickly establish integration between these disparate systems at a low cost of ownership. Designed for Systems Integrators (SI), Independent Software Vendors (ISV), Valued Added Resellers (VAR) and their customers, the **FusionWare Integration Server** enables users to integrate existing systems and rapidly build service-oriented applications without having to invest in .Net, Java or XML experts or add additional infrastructure and complexity to their IT environments.

Guided Intelligence (www.bi3.net), a new breed of Application Service Provider (ASP), is seeing its own return on investment using FusionWare. At Guided Intelligence, the need for low-cost implementation within a stormy e-business environment led to the selection of the FusionWare Integration Server as the foundation for its Guided Intelligence Analytical Server (GIAS). "We are a business intelligence software consulting company," says Warren Richman, President of Guided Intelligence. "We started as an ASP to deliver Business Intelligence on-demand to the mid-market."

"Clearly, Web services and the whole on-demand framework is essential for what we need to do." States Richman. "The challenge is integrating financial data from various sources and providing it in a single balance sheet or income statement. We needed to move to a more automated and repetitive process and FusionWare allowed our analytical server to do that with its Extraction Transformation and Load (ETL) capabilities, workflow designer and powerful transformation engine."

Information integration is a compelling value proposition for companies managing disparate systems. However, the costs incurred in back-end integration work can quickly wipe out savings for organizations if they aren't prudent in their selection of an integration toolset.

FusionWare connects to any data source, including structured (databases) and unstructured (MS Office documents, email, print files) data, and EDI. It also connects to existing applications through COM, .NET and Java interfaces, and even connects securely to other Web services. Many other products require a robust middleware tier – consisting of separate Application and Web servers and databases. Other tools either wrap or recompile libraries of legacy code into Java or a. NET-friendly language. Such approaches, of course, are non-trivial tasks for any IT shop. FusionWare's Integration Server - with a mere three-megabyte footprint and no external dependencies - can be quickly deployed on any connected server and be put to work immediately.

Such rapid integration provides dividends for Solution Providers and customers seeking simple solutions to formerly complex issues.

About the Author

Joe McKendrick is a research consultant and author specializing in information technology and management trends.

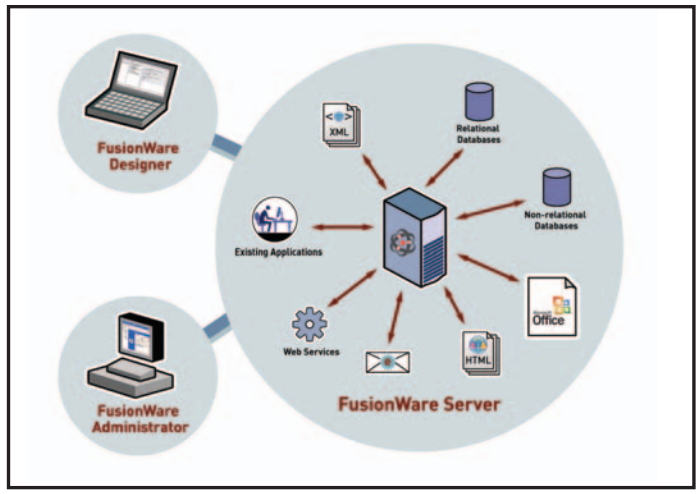


Figure 1: FusionWare Integration Server

FusionWare Corporation

Contact FusionWare at:
NA, Europe & South Africa
+ 1.949.250.4800
info@fusionware.net
Asia-Pacific
+ 61.2.9485.0200
info@fusionware.net.au

XML JOURNAL SUPPLEMENT

This Month

Defining Mainframe Transaction's Signature with an XML Schema
Edgardo Burin

Integrating mainframe applications into an SOA often carries the burden of dealing with metadata in the form of Cobol Copybooks. This metadata converted to an XML Schema format can be useful for a range of applications (from validation to creation of services).

David & Goliath
Ainhoa Serna & Jon Kepa Gerrikagoitia

Due to the great increase of data in XML format, companies increasingly have to face the issue of how to manage this data efficiently. To do this it is important to take advantage of XML's potential, and to integrate with applications that access data stored in relational database management systems (RDBMS).

Finding the Declarative Tipping Point
David Kershaw

Moving information from a database into an application may be the most common challenge developers face. How many of us make it through life without meeting object/relational (O/R) mapping in some form? Certainly not too many. Lately it has become equally difficult to avoid XML/relational (X/R) mapping

The Information Grid
Sandeepan Banerjee

Two somewhat contrary-sounding drivers fuel the emerging renaissance in enterprise data management – virtualization and convergence.

Defining Mainframe Transaction's Signature with an XML Schema

Converting Cobol metadata into an XML Schema using regular expressions processing

XML-Based Interop, Close up

In addition to the strategy side of Web services, there is also the protocol-oriented side of things, the XML side. Embracing not only XML itself but also the full range of mainstream XML-based technologies like XPath, XSLT, XML Schema, and SOAP. *XML Journal* has been delivering insightful articles to the world of developers and development managers since the year 2000.

It is our privilege to bring XML-Journal directly to readers of Web Services Journal, and vice versa. Anyone already familiar with the Web services world of SOAP, UDDI, and WSDL will find here articles and features each month that will interest them – about the cutting-edge technologies and latest products that are changing not only our industry, but the way the world exchanges information. To make it easy for you to find your way around, we have four distinct sections:

Content Management:
Organization, dissemination, and presentation of information

Data Management:
Storage, transformation, representation, and general use of structured and unstructured data

Enterprise Solutions:
Systems and applications that manage mission-critical functions in the enterprise

XML Labs:
Product reviews, book reviews, tutorials, and standards analysis

Defining Mainframe Transaction's Signature with an XML Schema

Converting Cobol metadata into an XML Schema using regular expressions processing

WRITTEN BY
Edgardo Burin

Integrating mainframe applications into an SOA often carries the burden of dealing with metadata in the form of Cobol Copybooks. This metadata converted to an XML Schema format can be useful for a range of applications (from validation to creation of services). This article explains how to automate the conversion from Copybooks to XML Schema using regular expression logic.

Cobol Copybooks 101

Mainframe metadata is usually defined using a subset of the Cobol language. Mainframe developers call these descriptions Copybooks. Cobol data definition is based on a hierarchical structure composed by two different types of items: Elementary Items and Group Items.

Elementary Item is the name Cobol assigns to a data item that is not further subdivided (analogous to variables in other languages). Elementary Items are composed of: a Level Number, a Data Name, and a Picture Clause. The Picture Clause (or PIC) allows us to declare the data format of the item.

In Cobol there are three basic data types: Alphanumeric (text strings), Numeric, and Alphabetic. Each of these formats is defined using a declaration sentence associated with a Picture Clause. The basic symbols used in the Picture Clause are: X for Alphanumeric, 9 for Numeric, and A for Alphabetic. The number of positions taken up by the data item is defined with a number inside parentheses, as in PIC X(10), which means an alphanumeric composed of 10 characters. There are more symbols and variants of declarations, but for the sake of simplicity I will restrict the explanation to these basic formats. For more details see the References section at the end of the article.

Group Items allow grouping a set of Elementary Items (or other Group Items) together. Group Items are composed of a Level Number and a Data Name, but don't contain a picture format. The Level Number creates a kind of hierarchical structure where one level groups all of the lower levels inside. The Level Number represents here the relationship that exists between different items in the definition.

For example, the following declaration:

```
01 COURSES.
 02 COURSE-ID.
  03 COURSE-TYPE   PIC X(3).
  03 COURSE-NUMBER   PIC 9(5).
 02 COURSE-NAME    PIC X(20).
```

represents a data definition composed of a Group Item called COURSES containing information about training courses. This group includes two items: the first is an Elementary Item called COURSE-NAME that is defined as a 20-positions alphanumeric field, and a Group Item called COURSE-ID. This group is composed of two Elementary Items: a three-character item called COURSE-TYPE and a five-position numeric item called COURSE-NUMBER. For a full description of the copybook see Listing 2.

Usually Level Numbers between 1 and 49 are free to use without restrictions. Levels don't need to be contiguous between them (a 01 group item can group several 04, 03, and 02 items). Levels 66, 77, and 88 have some special meaning assigned.

Since the main purpose of this article is to present a technique to convert from Cobol data definition into XML Schema, I will restrict the Copybooks to these basic formats (Elementary Items and Group Items), not including other kind of data (like arrays). In case of need the reader can extend the model to include other formats.

XML Schema 101

Having taken a look at the basics of Cobol data definition I will now move to our target: defining data structures in XML Schema. XML Schema allows us to construct valid XML documents. Schemas are defined using a vocabulary that names data items and their constraints (data types for example). The relationship between items is also part of the schema definition.

As I said before, XML Schemas allow describing the valid structure of a related XML file. Then, XML Schemas can be considered a metadata definition "from an underlying information

set," in the words of the W3C. The complete reference of XML Schema can be found in the W3C site (see the Reference section).

Elements are defined in the XML Schema with the element construct. Elements can be defined based on primitive datatypes or derived datatypes. Derived datatypes are defined using existing datatypes (primitive or not). XML Schemas allow us to define two type of elements: simpleTypes and complexTypes. For example a COURSE-ID can be defined as a complexType as in:

```
<element name="COURSE-ID"><complexType><sequence>
    <element ref="COURSE-TYPE"/>
    <element ref="COURSE-NUMBER"/>
</sequence></complexType></element>
```

This means COURSE-ID is a complex construct that includes a sequence of two other elements: COURSE-TYPE and COURSE-NUMBER. The sequence tag implies that the elements come in the order defined and without repetition. The ref attribute allows me to reference a type defined elsewhere. In this case, I will need to define a COURSE-TYPE and a COURSE-NUMBER datatype in the same Schema:

```
<element name="COURSE-TYPE"><simpleType><restriction base="string">
<length value="3"/><restriction></simpleType></element>
```

The element is a simple type defined based in the XML Schema primitive datatype string. I included some additional constraints (called facets in XML Schema language) using the length keyword. This definition means that I will allow just a string with a length of three characters. I used a primitive datatype string to define my simpleType. This primitive datatype is built-in to the XML Schema recommendation and includes for example string, Boolean, decimal, float, and double.

Additionally a numeric datatype can be defined using a similar statement as in:

```
<element name="COURSE-NUMBER"><simpleType><restriction
  base="positiveInteger">
<totalDigits value="4"/><restriction></simpleType></element>
```

Here I used another facet called totalDigits to constrain the numeric values. Also note that positiveInteger is a derived built-in datatype. Some examples of derived built-in datatypes are: normalizedString, integer, positiveInteger, and negativeInteger.

In a nutshell, we can define the XML Schema using primitive data types and derived data types defined using primitive or other derived data types. The primitive data types can be of any of the standard formats (for our application we will use just string and integer).

Simple datatypes are declared with the <simpleType> element and include the following basic attributes: name, base type, and they can contain a valid constraining facet. Complex datatypes are declared with the <complexType> element and they are defined by extension or restriction based on other datatypes.

Instead of referring a datatype defined in another portion of the same schema, derived data types can also nest datatype definitions, one inside the other as in:

```
<element name="COURSES"><complexType><sequence>
  <element name="COURSE-ID"><complexType><sequence>
    <element name="COURSE-TYPE"><simpleType><restriction
      base="string">
        <length value="04"/></restriction></simpleType></element>
        <element name="SERV-LENGTH"><simpleType><restriction
          base="integer">
        <totalDigits value="05"/></restriction></simpleType></
          element>
    </sequence></complexType></element>
</sequence></complexType></element>
```

Even when this kind of nested definition is less clear than the ones that use references, it will be useful for automating the generation of the XML Schema from the copybook as we will see soon. For a full description of the XML Schema obtained from the Cobol copybook see Listing 3.

Regular Expressions 101

In order to convert from Cobol to XML Schema we need to recognize certain patterns. For example we can build a rule saying that each group item in Cobol will correspond to a complexType in the schema, or that each elementary item containing a PIC clause will correspond to a simpleType. A useful artefact to recognize patterns in a text file is called a regular expression.

Regular expressions, called also regex, are used in several UNIX utilities and languages (Perl, awk, etc.). Regex allows us to locate a specific pattern or a particular sequence of characters in a string. This combination of characters is defined using a rather powerful syntax.

> "Because many integration tools have XML Schema as a way to define metadata or validate XML data, representing Cobol metadata as schemas can be applied in multiple scenarios"

Regular expressions are built around the use of special characters that are matched against the actual string. These special characters allow us to create a template against which each portion of the compared text is matched and processed in a certain mode.

For example, the regular expression ^.PIC * will match a string starting (^.) with just one character followed by the string "PIC" and followed by 0 or more blanks (will match APIC, BPIC__, but will not match CCPIC – two characters before PIC- or PIC – no character before PIC-). As seen in this example, special characters play an essential role in regex definitions. The Table 1 introduces the most common special characters used in regex.

Even when this is a very basic list of special characters it will suffice for our project. For a more extended information about regular expressions see the reference section.

The Project

In order to convert a copybook into an XML Schema I defined some rules of conversion. To simplify the scope of this project I will leave out some Cobol artefacts such as arrays, and I will centralize my attention on the basic structure of the Cobol metadata. For homework you can try

Special	Usage
.	Matches any single character.
*	Matches 0 or more occurrences of the preceding regular expression or character.
+	Matches 1 or more occurrences of the preceding regular expression or character.
?	Matches 0 or 1 occurrences of the preceding regular expression or character.
[...]	Matches any one of the class of characters enclosed between the brackets. A hyphen (-) is used to indicate a range of characters. All other metacharacters lose their meaning when specified as members of a class.
^	First character of regular expression, matches the beginning of the line.
$	Last character of regular expression, matches the end of the line.
\	Escapes the special character that follows.
|	Specifies that either the preceding or following regular expression can be matched (alternation).
()	Groups regular expressions.

Table 1 • Common special characters used in regular expressions

afterwards to extend the code in order to include these structures.

As said before, Cobol organizes the metadata in levels. To produce an XML Schema representation I will convert any level not including a PIC clause (that is any level that doesn't define a basic field) in a complexType. As one level usually includes other levels nested inside, I will nest the complexType definitions to mimic the Cobol definition, using the syntax seen in the XML Schema section.

The corollary of this rule is that any definition including a PIC clause will be considered a simpleType. We will use the length as a restriction in the definition of the field.

The Cobol example seen in the first paragraph:

```
01 COURSES.
 02 COURSE-ID.
  03 COURSE-TYPE   PIC X(3).
  03 COURSE-NUMBER   PIC 9(5).
 02 COURSE-NAME    PIC X(20).
```

can be translated then, as a complexType called COURSES that is composed of one complexType COURSE-ID and a simpleType COURSE-NAME. COURSE-ID is composed, in turn, of two simpleType fields: COURSE-TYPE and COURSE-NUMBER.

So with these two simple rules I can try to produce the schema. Now I will explain the tool we will use to achieve this objective.

The Program

In order to automate the conversion of the XML Schema I coded a java program that uses regular expressions to do the job. The java program reads the file containing the copybook, matches record by record against a pattern defined by a regex, and then produces a schema definition in another file. Since the definitions are usually nested, we need to keep some track of levels opened in order to produce the closing tags (</complexType>, </element>, etc.).

The program uses a set of classes included in Jakarta (mainly under org.apache.oro.text). These classes give us the basic functionality to search based on regular expressions:

```
import org.apache.oro.text.awk.*;
import org.apache.oro.text.regex.MalformedPatternException;
import org.apache.oro.text.regex.Pattern;
```

The regex functionality is provided by the three classes: Pattern, AwkMatcher, and AwkCompile. AwkCompile allows compiling a regex as in:

```
Pattern pattern = compiler.compile("(\\sPIC)|(\\sVALUE)|
  (^ *$)|(\\sCOPY\\s)");
```

The compiled pattern can be used afterwards to match against a string (contained here in an irecord variable) using an AwkMatcher object:

```
matcher.contains(irecord,pattern)
```

This method returns a Boolean true if the regex is found in the string. To produce an XML Schema, I write the header first:

```
outputfile.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
outputfile.println("<schema xmlns=\"http://www.w3.org/2001/
  XMLSchema\"");
outputfile.println("targetNamespace=\http://www.ibm.com\"" +
 "xmlns:router=\"http://www.ibm.com\">");
```

That header defines items such as the namespace of the schema and the encoding we will use. Refer to the XML Schema tutorial to learn about these clauses.

Once the header has been defined, the program parses each line of the copybook and checks if it's a complexType (that is a group) or a simpleType (that is a record containing a PIC clause). This is done searching *pattern = compiler.compile("(\\sPIC)");* for the simpleType and any record not containing a PIC, VALUE, or blanks for the group. The reader can change this search pattern or add others in case the copybooks include other types of definitions. The second case is solved by searching for any record that does not contain any of (\\sPIC)|(\\sVALUE)|(^ *$)|(\\sCOPY\\s). The ^ *$ searches blank lines. The others search a blank or tab followed by PIC, VALUE, or COPY.

If a simpleType is detected I produce a schema definition with an element name equal to the name of the field in the copybook. The facet in the schema will depend on whether the field is alphanumeric or numeric (i.e., PIC X(n) or PIC 9(n)). In case of an alphanumeric string the clause will be:

```
<element name="name-of-the-field"><simpleType><restriction
  base="string">
<length value="value –of-the-pic-clause"></length></restriction>
  </simpleType></element>
```

In case of a numeric the clause will be:

```
<element name="name-of-the-field"><simpleType>
<restriction base="positiveInteger"><totalDigits value=
  "value-of-the-pic-clause">
</totalDigits></restriction></simpleType></element>
```

In case of complexType I will nest the groups according to levels (01 first, then 03, then 05 etc.). I keep an account of levels opened, and close any complexType opened before, if needed (with the </sequence></complexType></element> syntax). This is done by the close method: if (level<lastLevel) close(complexTypes,level); and afterwards by writing a line in the form of

```
<element name="group-name"><complexType><sequence>.
```

This way, with a simple logic we can produce quite complex XML Schema nesting groups. For a full description of the java program see Listing 1.

An Example of Use

I designed the tool to produce a validation schema inside webMethods. webMethods is an Enterprise Integration environment that provides, as part of it functionality, a set of facilities for integrating mainframe applications and exposing them as services. I often need to validate complex data structures returned by the transactions invoked.

webMethods also allows importing an XML Schema to create a native webMethods Schema, which is useful for validating information that comes from the mainframe converted as XML data.

To show how to convert an XML Schema to a webMethods schema I will generate the XML Schema using the Cobol copybook (courses.cbl) seen before. First I will execute the java class using

```
java -cp .;jakarta-oro-2.0.8.jar CobolConverter courses.
  cbl courses.xsd
```

The schema generated (courses.xsd) can then be imported to a webMethods schema using a tool called webMethods Developer. Figure 1 shows a webMethods schema representation of the Cobol copybook imported.

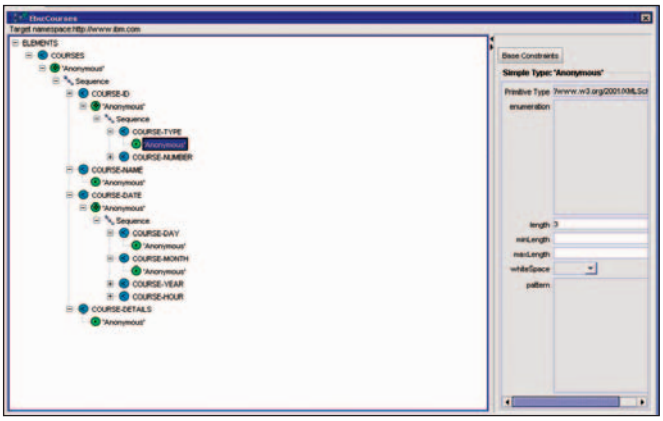


Figure 1 • webMethods Schema

Because many integration tools have XML Schema as a way to define metadata or validate XML data, representing Cobol metadata as schemas can be applied in multiple scenarios.

References

- For Cobol programming and data definition see *Advanced Cobol for Structured and Object-Oriented programming*. Deward G.Brown. John Wiley & Sons Inc.
- XML Schema Part 0: Primer, XML Schema Part 1: Structures, and XML Schema Part 2: Datatypes: www.w3.org
- *Mastering Regular Expressions*. Jeffrey E. F. Friedl, O'Reilly.

Author Bio

Edgardo Burin works for ING Canada as a solution architect in integration projects using webMethods. He works in different projects integrating mainframe transactions, MQ services, and Oracle databases using webMethods. He has more than 10 years of experience managing infrastructure. His areas of expertise are in Oracle databases, integration, and service-oriented architecture.

emburin@yahoo.com

Listing 1: Java code for CobolConverter.java

```
import java.io.*;
import org.apache.oro.text.awk.*;
import org.apache.oro.text.regex.MalformedPatternExcept
ion;
import org.apache.oro.text.regex.Pattern;

public class CobolConverter
{
 static PrintWriter outputfile = null;

   public static void main (String[] args) {

  String irecord = null; String recordTrim = null;
  AwkCompiler compiler = new AwkCompiler();
  AwkMatcher matcher = new AwkMatcher();
  Pattern pattern;
  int level=0, lastLevel=0;
  String[] line=null;
  int[] complexTypes= new int[100];
  try {
 BufferedReader buffr = new BufferedReader(new
FileReader(new File(args[0])));
 Outputfile = new PrintWriter(new BufferedWriter(new
FileWriter(new File(args[1]))));
   outputfile.println("<?xml version=\"1.0\" encoding=\
"UTF-8\"?>");
   outputfile.println("<schema xmlns=\"http://www.
w3.org/2001/XMLSchema\"");
 outputfile.println("targetNamespace=\http://www.ibm.
com\"" +
  "xmlns:router=\"http://www.ibm.com\">");

   while ( (irecord=buffr.readLine()) != null) {
    pattern = compiler.compile("(\\sPIC)|(\\sVALUE)|(^
*$)|(\\sCOPY\\s)");
    recordTrim = irecord.trim();

    // records without PIC map to complexType attri-
butes in the XSD
    // there is some logic also to close </ types
opened before with levels lower
 // if I find a level n I need to close the levels n+1
opened before that are
  // nested with a complexType opened. To do that I
keep the levels in an array
  // with an index for each level and
    // the content equal to the actual open complex-
Types
```

```
    if ((!matcher.contains(irecord,pattern)) && (ire-
cord.length()>0)){
     line = recordTrim.split("\\s+|\\.+");
     level = Integer.parseInt(line[0]);
     if (level<lastLevel) close(complexTypes,level);
     outputfile.println("<element name=\""+line[1]+"\
"><complexType><sequence>");
     lastLevel = level; complexTypes[level]=complexType
s[level]+1;
    } // if not PIC...

  // records with PIC map to elements the name is the
name of the field andthere
  // is a restriction based in the format X for string
9 for decimal and a
// length
  // according to the field length. The split function
allows to split based in
  // spaces or dots or parenthesis that spares many
instructions
    pattern = compiler.compile("(\\sPIC)");
    if ((matcher.contains(irecord,pattern)) && (ire-
cord.length()>0)){
     line = recordTrim.split("\\s+|\\.+|\\(|\\)");
     level = Integer.parseInt(line[0]);
     if (level<lastLevel) close(complexTypes,level); //
close open levels
     if (line[3].equals("X")) // case alphanumeric
  outputfile.println("<element name=\""+line[1]+"\
"><simpleType>" +
  "<restriction base=\"string\"><length " +
  "value=\""+line[4]+"\"></length></restriction></sim-
pleType></element>");
     if (line[3].equals("9")) // case numeric
      outputfile.println(
        "<element name=\""+line[1]+"\
"><simpleType><restriction " +
    "base=\"positiveInteger\"><totalDigits " +
 "value=\""+line[4]+"\"></totalDigits></restriction></
simpleType></element>");
     lastLevel = level;
    } // if PIC...

   } // while...
   close(complexTypes,-1);
   outputfile.println("</schema>");
      outputfile.close(); // close output file
  } catch (MalformedPatternException e) {
   System.out.println("IOException processing file.."+
e.toString()+"\n");
  } catch (Exception e) {
   System.out.println("IOException processing file.."+
e.toString()+"\n");
  }
 } // main...

 /*
  * close -- close all the previously open xml schema
tags  with a level bigger than
  *  the actual level
  */
 private static void close(int[] complexTypes, int
level) {
     for (int i=0; i<complexTypes.length; i++) {
      if ((i >= level) && (complexTypes[i] > 0)) {
       outputfile.println("</sequence></complexType></
element>");
       complexTypes[i]--;
      } // if
     } // for...
   } // close...


} // class
```

Listing 2: Cobol copybook

```
01 COURSES.
 02 COURSE-ID.
  03 COURSE-TYPE  PIC X(3).
  03COURSE-NUMBER PIC 9(5).
   02 COURSE-NAME         PIC X(20).
 02 COURSE-DATE.
  03 COURSE-DAY   PIC 9(2).
  03 COURSE-MONTH PIC 9(2).
  03 COURSE-YEAR  PIC 9(4).
  03 COURSE-HOUR  PIC 9(2).
 02 COURSE-DETAILS        PIC X(30).
```

Listing 3: XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:
router="http://www.ibm.com">
<element name="COURSES"><complexType><sequence>
 <element name="COURSE-ID"><complexType><sequence>
  <element name="COURSE-TYPE"><simpleType><restriction
base="string">
  <length value="3"></length></restriction></simple-
Type></element>
  <element name="COURSE-NUMBER"><simpleType>
  <restriction base="positiveInteger"><totalDigits
value="5">
  </totalDigits></restriction></simpleType></element>
 </sequence></complexType></element>
 <element name="COURSE-NAME"><simpleType><restriction
base="string">
 <length value="20"></length></restriction></simple-
Type></element>
 <element name="COURSE-DATE"><complexType><sequence>
  <element name="COURSE-DAY"><simpleType><restriction
base="positiveInteger">
  <totalDigits value="2"></totalDigits></restriction></
simpleType></element>
  <element name="COURSE-MONTH"><simpleType>
  <restriction base="positiveInteger"><totalDigits
value="2">
  </totalDigits></restriction></simpleType></element>
  <element name="COURSE-YEAR"><simpleType><restriction
base="positiveInteger">
  <totalDigits value="4"></totalDigits></restriction></
simpleType></element>
  <element name="COURSE-HOUR"><simpleType><restriction
base="positiveInteger">
  <totalDigits value="2"></totalDigits></restriction></
simpleType></element>
 </sequence></complexType></element>
 <element name="COURSE-DETAILS"><simpleType><restrictio
n base="string">
 <length value="30"></length></restriction></simple-
Type></element>
</sequence></complexType></element>
</schema>
```

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

Offering leading software, services, and hardware vendors an opportunity to ***speak to over 100,000 purchasing decision makers*** *about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management*

Don't Miss Your Opportunity to Be a Part of the Next Issue!

Get Listed as a Top 20* Solutions Provider

For Advertising Details Call 201 802-3021 Today!

*ONLY 20 ADVERTISERS WILL BE DISPLAYED. FIRST COME FIRST SERVE.

SYS-CON MEDIA

The World's Leading i-Technology Publisher


WRITTEN BY **AINHOA SERNA & JON KEPA GERRIKAGOITIA**

David & Goliath

A comparison of XML-enabled and native XML data management techniques

Due to the great increase of data in XML format, companies increasingly have to face the issue of how to manage this data efficiently. To do this it is important to take advantage of XML's potential, and to integrate with applications that access data stored in relational database management systems (RDBMS).

Currently, there are no available tools that can be used to analyze whether XML data is best stored in conventional RDBMSs or in native XML database systems, in particular applications involving legacy or new systems. The most appropriate solution will depend on the process type (read only or read-write), the XML document's structure (single or multiple hierarchies, or dynamic structure changes), data volume, and the application priorities (fast queries, concurrent updates), for example.

In this article we present, in a graphical form, the response times for some common operations using SQL and XPath. We store XML data using different storage and management solutions: Oracle ("Goliath") – an XML-enabled database, and eXist ("David") – a native XML database (NXD). eXist is a very recent DB that needs maturing but presents many virtues; "Goliath" is Oracle, a mature, robust, secure, and reliable RDBMS. We're bringing these two databases face-to-face like "big titans" to demonstrate which is the best at different development scenarios.

The main advantage that we have found for the eXist NXD is the direct XML management it offers as a native XML database. Another advantage is that eXist is open-source software, which also provides all of the requisite libraries. The means that it offers an inexpensive technology option, since only training costs will be incurred in its use.

In XML data management, companies wonder how to manage XML data and integrate with legacy RDBMS. Database queries must comply with the same structure as XML data. The access to RDBMS is direct through SQL sentences. NXD allows access to XML data directly. Figure 1 shows the three-tier architecture used in this experiment.

XML Data Management

In this article we investigate two different approaches to XML data management: one that uses conventional database techniques to deal with XML formatted data (often called XML-enabled database systems) and an alternative that uses XML formats as its basic storage format, native XML database systems. For the fist approach, XML-enabled databases, we used Oracle, a widely used commercial DBMS. For the second approach, we used eXist, an open source native XML database.

Oracle (XML-enabled)

Oracle (and other major RDBMS vendors) has extended its basic RDBMS functionality to include XML data management capabilities. This kind of database is named XML-enabled. By using XML-enabled databases we can transfer data from relational tables to XML documents in both ways.

For using Oracle XML-enabled database there are two main solutions:

1. The whole XML document is stored in XMLType column:CLOB [8] (XML document stored as CLOB, character large object)
2. An XML document's structure is mapped to database columns (XML SQL Utility).

eXist (native XML database)

Why store XML documents somewhere that doesn't have XML structure (RDBMS)? Why adapt the XML structure

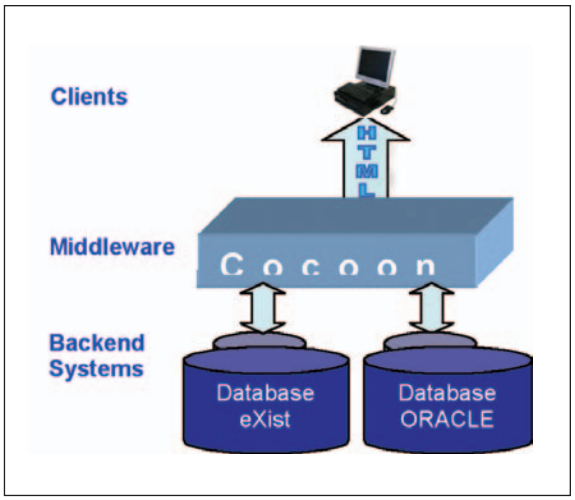


Figure 1 • Cocoon three-tier architecture

AUTHOR BIO

Ainhoa Serna is lecturer in the department of Informatics at ETEO (Mondragon Unibertsitatea). She has also worked as Project Leader at engineering and Research projects since 1995. Her research work is focused on XML.

Jon Kepa Gerrikagoitia is lecturer in the department of Informatics at ETEO (Mondragon Unibertsitatea). He has participated in a European Research Project and engineering projects in the last 10 years. His research work is focused on XML Web Services.

to that kind of an RDBMS when there are more natural ways to store XML documents? Open source eXist NXD will be the best solution in certain cases.

3. Open-source eXist, Native XML Database (NXD)

Let's analyze each solution's pros and cons.

XMLType:CLOB

The Oracle XMLType:CLOB stores the whole document, which is good because we don't loose context or data hierarchy, and XPath's queries are allowed. In addition, it allows us to use the XPath syntax to update specific elements and attributes without rewriting the document. This solution is not as good as object-relational solution querying and updating data. Later, we will present the response times using XMLType query engine.

In previous database versions (Oracle 8i, for example) we had no XMLType, we only had CLOB choice, and there was no way to update an XML document's part. It was necessary to change the whole document and XPath's queries were not allowed. Using more recent Oracle RDBMS versions such as 9i and 10g, this kind of XML management is much better supported. Also, this has become the most common solution for unstructured XML documents (document-centric).

XMLType has predefined member functions to extract XML nodes and fragments. We use various functions such as extract( ) (this uses XPath to return fragments as XMLType), and getStringVal( ) (which returns a document as a string). These functions are embedded into SQL sentences to provide XML functionality. For example:

Create table:

```
Create table clientes of xmltype;
```

Insert document into a table:

```
Insert into clientes values ('<?xml…..');
```

Update:

```
update clientes set object_value =
  updatexml(object_value, '//nombre/
  text()', 'Ainhoa');
```

Query with XPath:

```
Select c.extract('/cliente').getStringVal()
  from clientes c;
```

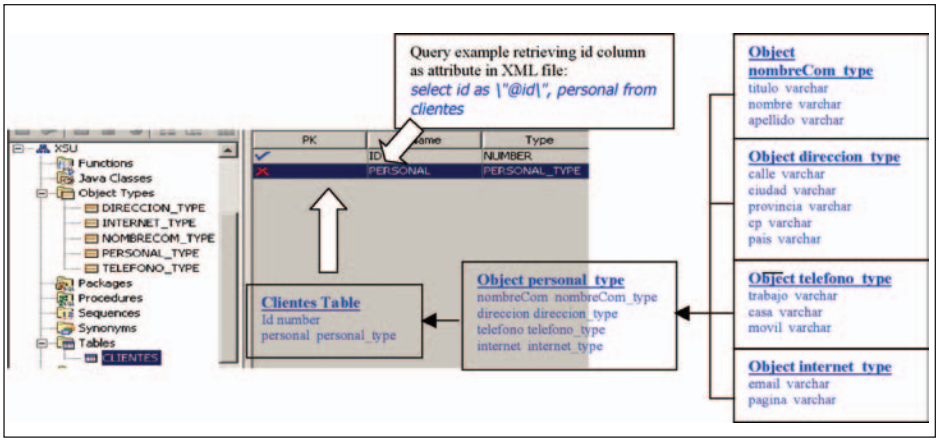


Figure 2 • Object-relational table stored

Query with several tables:

```
select extract(c.object_value, '/cliente').
  getStringval() "clientes", extract(cu.
  object_value, '/cuenta').getStringval()
  "cuentas" from clientes c, cuentas cu
  where extractvalue(cu.object_value, '/
  cuenta@cliente-id') = extractvalue(c.
  object_value, '/cliente@id');
```

Eliminate:

```
delete from clientes where
  extractvalue(object_value,'/
  cliente@id')='0011010';
```

The XMLType also allows structured data to be stored as objects. This is similar to XSU object stores, and for this reason we do not consider it further here.

XSU

The XML SQL Utility is a set of APIs that support operations between XML documents and relational databases. It is possible to build an XML document from an SQL query, and to insert, update, and delete a relational data table from an XML document. To store XML documents in relational tables it is necessary to map the document nodes to a relational table format. So, XML-enablement allows getting XML documents from SQL queries (no new knowledge required) that can be formatted with XSL to any client application format.

Sometimes however, it is not so easy to build an RDBMS structure that matches an XML document. This is because XML documents are hierarchical and may have a dynamic structure that can change easily. In this case, we would need to rebuild the XML document with its structure each time we query the database.

Two storage and retrieval methods are possible for XSU: object-relational table storage (information stored in defined types), and relational table storage (data is stored in a relational table).

We implement both solutions to investigate their implications in code development, and the differences between structure creation and response times.

XSU does not support attribute nodes and element nodes with more than one occurrence in the same level. In the first case we should have to transform attributes in element nodes, for example <client id="001001"> would be <client> <id>001001</id>. In the second case, we would need to fragment the document, thus inserting the part into various tables. There would be an XSU instance to insert each document into its corresponding table. We use XSLT stylesheets to transform each document into the appropriate format before writing it to the database.

With XSU object-relational table storage, information is stored in defined types according to the structure shown in Figure 2.

- The table contains an object.
- The objects contain data.
- XML hierarchical structure is maintained.

Using XSU relational table storage information is stored in a relational table in accordance with the structure shown in Figure 3.

- The table contains data. The objects are empty.
- There is no any hierarchical structure. Structure is built when selected (using objects).

Using the XSU table solution, only one nesting level after root element is allowed. All elements are in the same level.

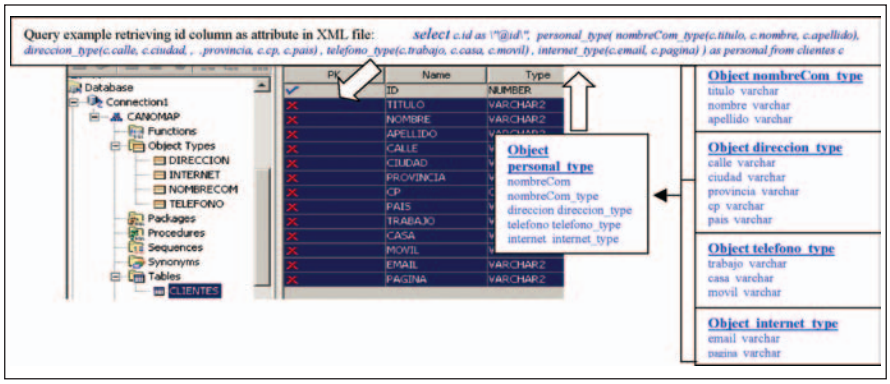


Figure 3 • Relational table stored

We can only represent one nesting level. Response time is elevated more than with the previous case. In the selected approach we have to rebuild the XML document's initial hierarchical structure, thereby creating the objects dynamically with data from the table.

If we have the structure stored in objects it is easier to retrieve XML documents in their initial hierarchy format. XSU (XML-SQL Utility) give us the following functionalities.

- It converts a relational query to XML. Column names would be element names in the document; by default ROW is the element root and there are as many ROW elements as there are rows needed to retrieve the query. These ROW and ROWSET names can be customized through XSU methods respectively (setRowTag(String tagname) and setRowsetTag(String tagname)).
- When using XML documents it is possible to insert, update, and delete data from relational tables according to certain rules. For example, "SELECT * FROM CLIENTE" in XML would be:

```
<?xml version = '1.0'?>
<ROWSET>
   <ROW num="1">
      <ID>001</ID>
      <NOMBRE>Pablo</NOMBRE>
      <APELLIDO>Nocedal Renovales</
APELLIDO>
      <CIUDAD>Sopuerta</CIUDAD>
      ........
      <EMAIL>pnocedal@eteo.mondragon.
edu</EMAIL>
   </ROW>
  <!-- filas adicionales ... -->
</ROWSET>
```

We have used Java as a programming language. XSU needs JDBC connectivity interfaces and Oracle's XML parser. The following classes provide this functionality.

```
OracleXMLQuery qry = new
OracleXMLQuery(conn, "select *
  from cliente");
```

This class allows us to retrieve the XML content from the query:

```
String str = qry.getXMLString();
```

To store XML documents in a relational database:

```
OracleXMLSave sav = new
  OracleXMLSave(conn, "cliente");
```

We indicate JDBC connection details and a working table in the constructor method. Next we will use the methods:

```
insertXML(fichero_xml)
updateXML(fichero_xml)
deleteXML(fcihero_xml)
```

With the insertXML method we insert as many rows in the table as element ROW has in the XML file. The unique restriction is that element names and table column names have to match.

To use updateXML it is necessary to indicate the key values to be updated, as well as new values. The delete process is quite similar. The following is an example of an update with an XML file:

```
OracleXMLSave save = new
OracleXMLSave(conn, tabla);
save.setKeyColumnList(keys);
save.setUpdateColumnList(campos);
save.setIgnoreCase(true);
save.updateXML(new java.net.URL(
  fichero_xml));
```

This code would be the same as: UPDATE CLIENTE SET TITULO=?, NOMBRE=?, APELLIDO=? WHERE ID =? We use the Jaxen XPath engine, because it has been selected to represent a conventional top-down, tree-traversal–based query engine. For our experiment Jaxen runs on DOM implementation.

```
XPath xp = new DOMXPath("//nombreCom");
xp.evaluate(docDOM);
```

eXist

eXist is the native XML database (NXD) used in our investigation. NXD databases store and manage XML documents as well as XML model's components (DTD, Schema, XSL). XML data is stored as-is; no conversion is needed. The logical storage unit is an XML document, so the input and output unit will be an XML document too. Thus, an NXD database engine accesses XML data with no performance lost because there is no conversion involved in this process.

Our investigation has used eXist NXD because it fulfilled the following requirements: it is written in JAVA, it is open source, it is platform independent, the search engine is optimized compared to other NXDs such

	PC features
Processor	AMD Athlon (tm) XP 2800+
Memory	1 GB
Operating System	Windows 2000
Software	Web container: jakarta-tomcat-4.0.4 (Apache)Version Java (SUN): j2sdk1.4.1_01

Table 1 • Tested PC features

as XIndice, and it has support for document and node updates. Indexes for elements, attributes, and keywords are organized by collections instead of by document (which improves searches for complete collections).

The experiment we conducted is developed using XSP technology with its XSP logicsheet taglibs. XSP technology is very appropriate for generating dynamic content for people with no programming skills. With XSP we separate presentation and logic. We have two collections: customers and accounts. In each collection XML documents are stores with associated data.

We created an initial collection with 5,000 XML documents. Each document is 592 bytes. Next, we created four new subcollections with the same number of files (5,000). Finally, we added two new subcollections of 4,000 and 6,000 XML documents, respectively. We increased the CLIENTS collection to 30 MB. The experiment processed the same set of queries with increasing data volumes to test the scalability of exist.

Performance

Table 1 shows PC features where tests have been assessed.

Types of Tests

We compare query execution times for eXist, Oracle (with XSU-SQL and XPath-Jaxen, an external XPath query engine which is based on a conventional tree-transversal–based approach). Table 2 shows the different storage solutions investigated. Tests are grouped in:

XML:DB XPath
XML:DB test
XML:DB robustness

For the first group (XML:DB XPath), different query types have been used to obtain performance conclusions. We conducted tests using the following four criteria.

1. Recurring searches (//node, full scan) to compare with path-qualified searches
2. Search for a concrete attribute node with a concrete value with recursive search and path-qualified search
3. Searches with two conditions were evaluated to test performance
4. Predefined XPath functions such as "contains"

In the second test group (XML:DB test), we tested maintenance tasks as:

- Add/delete/update documents
- Create/drop collections

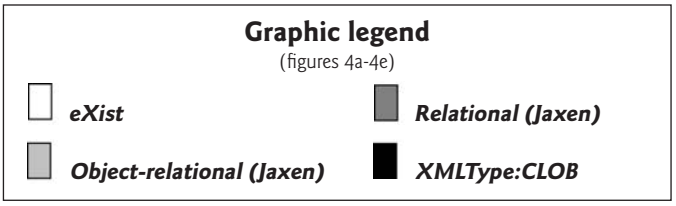
XUpdate for a concrete attribute node "ID" (Figure 4(a)).

We processed the test with increasing data volumes. The last group (XML:DB robustness) was done to test the maximum collection capacity.

XML:DB XPath

Tested queries are the most frequently asked by users, for example, data related to a customer's ID. We have grouped query tests as below, showing time averages for different data volumes.

- Recurring searches (//node, full scan), element node (Figure 4(b) *"//nombreCom"*)
- Search for a concrete attribute node "ID," with a concrete value with recursive search (Figure 4(c) *"//cliente[@id= "0011001"]//nombreCom"*)
- Search for a concrete attribute node "ID," with a concrete value with path-qualified search (Figure 4(d) "*/cliente[@id="0011001"]/personal/nombreCom*")
- Recurring searches with predefined XPath functions such as "contains()," (Fig. 4(e) «*//nombre[contains(.,»Jon11001»)]*»)



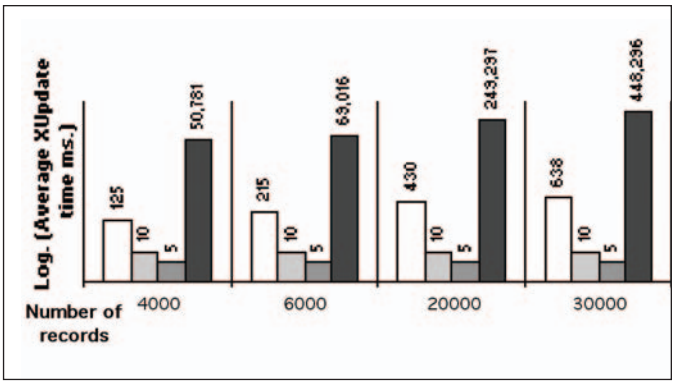


Figure 4a • XUpdate

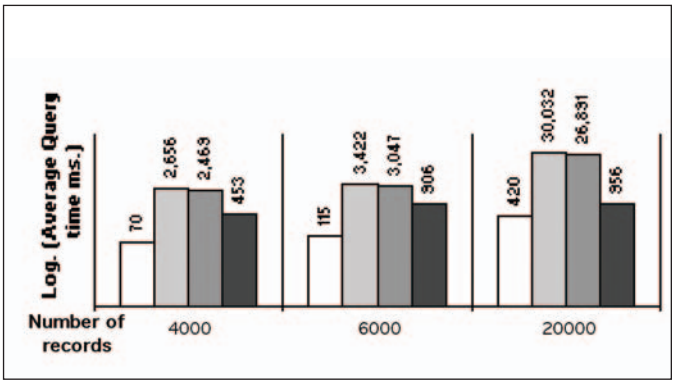


Figure 4b • Query1 «//nombreCom

Experiment Results

The eXist search engine is optimized for recurring searches (// symbol); for this reason the response times are good, as can be seen in Figures 4(b) and 4(c). eXist has the best times in all of the searches.

Using XPath standard functions like "contains" (see Figure 4(e)), the search engine needs a complete scan of each node of the current context node, and this results in longer response times than those using the "=" operator.

Query execution times increase at least linearly with increasing source data size. We

ORACLE			eXist
XSU object stored	XSU table stored	XMLType:CLOB stored	Native stored
query SQL		Query SQL-XPath- extract()	XPath
query XPath Jaxen			

Table 2 • Storage solutions

Time msec	eXist	Object-relational table	Relational table	XMLType: CLOB
0-39	97.71%	99.49%	99.80%	99.90%
0-15	66%	82%	83%	69%

Table 3 • Experiment results

show linear scalability of eXist's indexing, storage, and querying architecture. eXist's insert operation times are good, but object-relational and relational table in Oracle are better. Inserting 30,000 XML documents produced the results shown in Table 3.

Insert operation times are very similar in XMLType:CLOB and eXist. These times are worse than object-relational tables and relational tables (we don't consider transformation to adequate format time with Oracle XSU storage).

The random access time in XMLType: CLOB is worse than other solutions. However the full scan (search for all nodes //nombreCom) performance time is good, but with Jaxen (object-relational table and relational table), recurring searches times are better than XMLType: CLOB times.

The results show that the transactions (update, select) against the structured storage model are much faster than the CLOB-based storage model, as expected.

The Oracle9i XMLType enables querying collections of XML documents through the extract function, which takes an XPath parameter. When a query is run against XMLType:CLOB tables, all 30,000 records have to be brought into the memory, parsed, and examined. The performance of Oracle XMLDB is acceptable.

NXD's disadvantages: indexes and data sizes are much larger than XML-enabled database. XUPDATE operation times are shown in Figure 4(a), and we clearly notice that XMLType:CLOB has very bad performance.

Why We Need NXD

The results show that eXist's query engine outperforms the tree-transversal–based approach (for example, Jaxen) by an order of magnitude.

Queries – to achieve a useful comparison of access times, we use Jaxen XPath engine with the XML SQL Utility (which does not have an XPath engine itself). This requires that we perform a two-step query process. First a set of intermediate results is obtained by performing a normal SQL query, and then we perform an XPath query over these intermediate results to obtain the final results. In this way, we were able to perform the XPath query in three different cases, as shown in Table 2.

Also, we see that in the case of the object-relational table, a query to reconstruct the original structure of the document is much simpler that the query needed to do the same in the case of the relational table (see Figures 2 and 3).

Scalability – schema (structure) evolution involved changes to the database schema. The XML data model is flexible to model documents, XML-aware full-text searches, and structured query languages like XQuery. In this way in an NXD solution, XML documents can be stored and queried in a single location, rather than multiple locations held together thanks to programming code (RDBMS solution with several tables storing an XML document with several hierarchical levels having multiple occurrences). This scenario shows more complex code in RDBMS than NXD, and so, it is the least able to deal with scalability issues.

Flexibility – NXD handling schema evolution is better than RDBMS implementation.

Updates – node-level updates are supported by both RDBMS and NXD, but NXD offers much better performance than the XMLType:CLOB solution, as shown in Figure 4(a). Object-relational table and relational table storage have the best update performance, but NXD performance is close to these. In a scenario with massive update tasks having critical response time, we should discard the XMLType:CLOB solution. Node-level updates reduce the cost of updating large documents and not all databases have this capability implemented; maybe they only have document-level updates implemented.

Not all NXD can scale into the gigabyte or terabyte; RDBMS clearly can. Content management systems are being built on native XML databases rather than on the file system or relational databases. XML content that is difficult to store relationally should be stored in an NXD, because it's

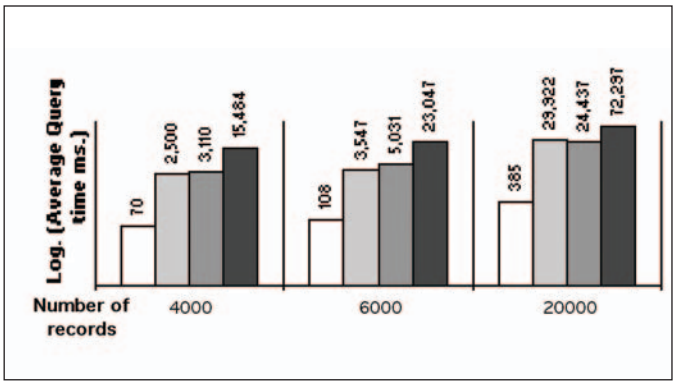


Figure 4c • Query2 "//cliente[@id="0011001"]//nombreCom"

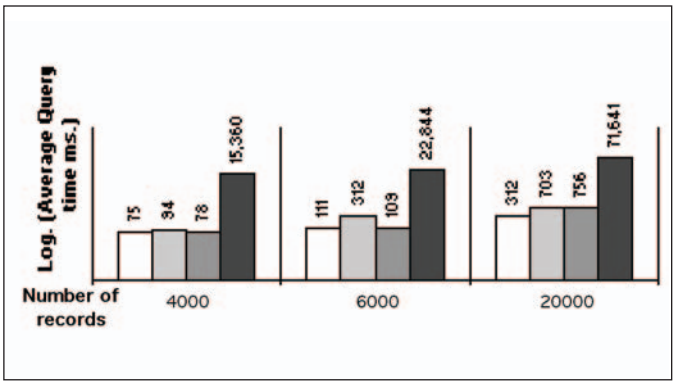


Figure 4d • Query3 «/cliente[@id=»0011001»]/personal/nombreCom

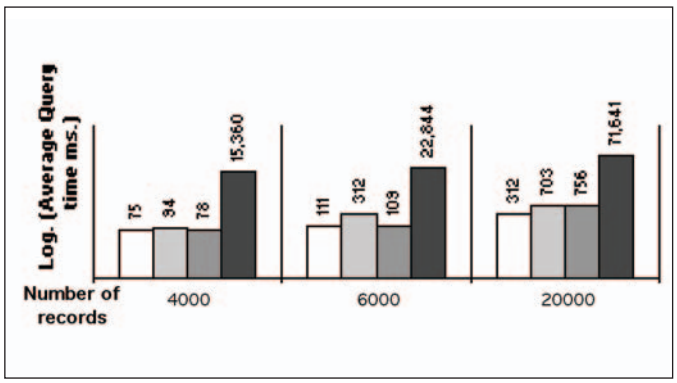


Figure 4e • Query4 "//nombre[contains(.,"Jon11001")]"

much more efficient, e.g., technical manuals that currently are built as hypertext documents with dynamic multimedia content.

NXD has not been designed to substitute RDBMS, which are known today to be very reliable and robust. Both NXD and RDBMS can work together. XML data can be stored in NXD and relational data models will remain stored in RDBMS.

In order to take advantage of both technologies, we recommend a combined solution. If data structures change frequently, NXD (eXist) is the best bet. If transactions are mainly queries, we will choose NXD (eXist) as the best solution again.

References

- Brogden, B. et al, Mark Gaither et al, 2003. *Cocoon 2 Programming Web Publishing with XML and Java*, Sybex, USA.
- Akmal B. Chaudhri, Awais Rashid, Roberto Zicari, 2003. *XML Data Management*, Addison Wesley.
- *XSP logicsheet (Extensible Server Page):* http://spqr.oucs.ox.ac.uk:8080/exist/xmldb.xsl.html
- *Cocoon (Apache Cocoon Project):* http://xml.apache.org/cocoon
- *Oracle (XML-enabled database):* www.oracle.com
- *eXist (Open source native XML database):* http://exist-db.org
- *XML (Extensible Mark up Language):* www.w3.org/XML
- *XSL (Extensible Stylesheet Language):* www.w3.org/Style/XSL
- *XSLT (XSL Transformations):* www.w3.org/TR/xslt
- *XMLType:CLOB (Oracle Corporation):* www.oracle.com/technology/oramag/oracle/03-jul/o43xml.html
- *XPath (XML Path Language):* www.w3.org/TR/xpath
- *XUpdate (XML:DB Initiative standard):* www.xmldb.org/xupdate
- *XIndice (Apache XML Database):* http://xml.apache.org/xindice

aserna@eteo.mondragon.edu

jkgerrik@eteo.mondragon.edu

WRITTEN BY **DAVID KERSHAW**

Finding the Declarative Tipping Point

XQuery, XML, and the RDBMS

Moving information from a database into an application may be the most common challenge developers face. How many of us make it through life without meeting object/relational (O/R) mapping in some form? Certainly not too many. Lately it has become equally difficult to avoid XML/relational (X/R) mapping. Because XML, and especially XML Schema (XSD), are object-like paradigms, the mapping difficulty is approximately the same. However, under the ever-expanding influence of XML, the extract, transform, load process that gets data from a database into an application (and vice versa) may be about to get radically more simple and declarative.

Today if you find yourself thing thinking about X/R mapping you have three main options to look at:

1. Bind Data Definition Language (DDL) to XSD (possibly creating the XSD in the process)
2. Use XSD to create or map to DDL
3. Use XQuery to extract and/or declare data bound, at least implicitly, to both XSD and DDL

It is a big field and there may be other approaches, but let's look at how these three options play out in the market for practical solutions and developer mindshare.

In practice, binding DDL to XSD happens regularly. Most often this involves developers using tools such as Altova XMLSpy to generate the XSD from the database structure, and from there using JAXB, Microsoft's Xsd.exe, XMLSpy, or another means to tie the XML structure to objects. This route, however, does not get you away from the need to somehow query the database and iterate through the results, inserting individual values into objects. There are other tools that can automate these steps, but the requirement still exists.

The second option of constructing the XSD first and then deriving both the classes and the DDL from it is very similar, at least in some ways. A main advantage of this approach over using the database as the primary structure is that you have a clearer single source for your data management and code structures, whereas if the database structure takes the lead, you typically have XSD or another type of model working as an intermediary to your classes. Moreover, XSD typically offers something very near to a superset of the typing capabilities of both the database and your programming language. This option is most often performed using XMLSpy to build the XSD and automatically convert it to a database structure, then optionally, using a visual data integration tool like Altova MapForce to map between individual fields, insert data processing rules, and generate customized data conversion code for use in your project.

The third option utilizes XQuery. As you are probably aware, XQuery is still a work in progress, but XQuery has been fairly stable for about a year and it is very near to becoming a formal W3C recommendation, as a peer of XPath 2.0 and XSLT 2.0. Moreover, XMLSpy, and perhaps other tools, now can be used for writing, debugging, and interpreting these new standards, and MapForce can even generate XSLT 2.0 and XQuery code from XML Schemas. If you are an application developer, I suggest you do not wait. Grab an implementation and take a long look at these new standards because they may just complete the enterprise mainstream's shift to declarative development.

What does all this mean? How will these new standards provide the missing link? Declarative development means not programming your application, but rather modeling and describing the processes and links between processes using XML technologies. XSLT and XSD are today's declarative entry points and are in wide use, but XSLT 1.0 has significant limitations as an application platform. XSLT 2.0 (along with XPath 2.0) was designed to address those limitations, however XSLT 2.0 is not intended as a query language. For a complete data access, transformation, delivery, and presentation solution, utilizing XSLT 2.0 (for visualization) plus XQuery (for data access) is the next logical step to take. Keep in mind that the custom-developed part of the typical business application is not algorithmically complex. Rather, that custom-developed core logic

AUTHOR BIO

David Kershaw is the Manager of Professional and Educational Services for Altova, Inc., where he manages the company's broad range of services, including business solution building, development project consulting, and online and on-site training. David is a frequent writer and speaker on XML, software development, and data integration issues, and has previously served as director of engineering for Classwell Learning Group and group director of engineering for Organic, Inc. He received his Masters from Harvard University and his Bachelors from the University of Massachusetts.

is really just a matter of modeling one or more complex processes that have complex data. The benefit of moving to declaring that core part of your application in XML, XQuery, and XSLT 2.0 is that more of the mundane code that still must exist somewhere becomes a commodity part of your application server infrastructure. The result is faster development, fewer quality issues, and greater long-term maintainability.

This revolution has admittedly been a long time coming. The changes responsible for boosting developer productivity will:

- End the need for O/R mapping frameworks
- Eliminate most custom code at the network interface and in the domain model
- Finally result in a universal object query language

Most programming model changes are evolutionary. This shift is revolutionary, not because it has anything to do with a switch to XML technologies, but for one simple reason: it is possible that for many business applications hosted on the typical corporate intranet (e.g., purchase order or personnel management applications) there soon won't be any programming necessary. This will free you up for less tedious tasks that require more creativity and a better understanding of your customer's requirements. The previous, often non-XML-based moves in this direction didn't come close to offering that outcome. Before slicing and dicing the three points above, let's take a look at how it all ties together, and how the new standards will push us past a declarative tipping point.

Imagine a typical server-side, browser-based human resources application for a large firm built in the near future. In one component of the application an HR professional can browse, edit, create, or delete employee information. That information is stored in a database that is used by several other applications and holds other types of related information. Without getting into all of the requirements, let's say that this application has ten main screens, a good number of business rules, and must create, read, update, and delete data from the database.

Suppose the architecture of this application is based on the J2EE frameworks (although the scenario is equally likely in the world of .NET). The presentation side of the equation will be constructed as XML from the standard JSP tags, XHTML, and CSS. Those JSPs will tend to be simple declarative pages that make calls against an EJB business API. The information returned will be in the form of XML. That XML will be transformed using XSLT 2.0 into XHTML. So far this is fairly typical, even for today.

On the business API side, there will be one or more stateless session EJBs. These are not today's custom developed EJBs – instead they are built into the application server. Their primary mission will be to host and execute an XQuery that is embedded as part of their XML descriptor and to return the XML result (or an error message in exceptional situations). On the JSP side, each JSP knows what operation it is responsible for triggering and displaying. A standard tag library will set up the EJB request and direct the resulting XML through an XSLT 2.0 stylesheet for formatting as HTML. That HTML will then be displayed as the complete page. Everything done in this future application is declarative, from the UI to the business API to the database manipulation, and no functionality will be sacrificed to get there.

"Grab an implementation and take a long look at these new standards because they may just complete the enterprise mainstream's shift to declarative development"

Follow that basic pattern of JSPs containing tags that call session beans that use XQuery to manipulate data enough times, add in a dash of XACML (XML Access Control Markup Language), XSLT 2.0, and other facilitating declarative infrastructure, and pretty soon you're talking about a real application. Can this exclusively XML-based pattern do everything that might be required of the application the way compiled code does today? For the majority of business intranet applications the answer is "yes." Between the power of XQuery, XPath 2.0, and XSLT 2.0, there isn't much you would need to fall back on Java (or C#) for.

What this means is that in one stroke the new standards knock off a pair of long-standing problems that have repeatedly failed to get full industry consensus in the past: O/R mapping and a common query language.

But wait, XQuery and the others provide an X/R solution, not an O/R one, right? That is basically true. However, if all of the objects you create for your application are generated from XSD or are involved in executing XQueries or XSLT 2.0 stylesheets, you don't really have any significant objects. In that case, they will be pushed down into the application development infrastructure to become a standard part of the application server. Now you don't have that O/R problem anymore – it becomes an X/R issue with a very clear solution, and the infrastructure is already moving in this direction.

What stands between the future and us? Given the way the client sides of so many enterprise Web applications are built today, getting the procedural code out of the picture becomes just a matter of connecting the persistence layer (XQuery) and the client (XSLT-HTML). That is where the declarative deployment descriptors of EJBs and Web services come into play. Add in an XSD-defined domain model, as well as some client-side JavaScript, if you must, and the transformation is virtually complete, at least for the most common ~80 percent of business applications.

With the shift to declarative application development you will see:

- Significantly increased interoperability due to the shift to technology's most open standards, i.e., the W3C's XML recommendations
- Much greater flexibility to adapt and configure applications because they are now boiled down to just the essential logic that can be more easily understood and quickly modified and validated within XMLSpy
- A new way of looking at systems development where coded algorithms are islands within an architectural ocean of infrastructure and declarative data transformation and flow

The upshot is that as a modern developer you need to get ready because more change is coming. Study your current design patterns and your XPath 2.0 (the foundation of both XQuery and XSLT 2.0). While you're waiting for your application server to catch up, check out the more recent code generating data integration tools such as MapForce that are helping to make this vision a reality. Finally, get comfortable with the idea of a declarative future.

david.kershaw@altova.com

WRITTEN BY **SANDEEPAN BANERJEE**

The Information Grid

XML and databases: moving toward convergence

Two somewhat contrary-sounding drivers fuel the emerging renaissance in enterprise data management – virtualization and convergence. Virtualization is a framework for dividing up the resources of an organization into multiple execution environments through the application of one or more technologies such as hardware clustering, software partitioning, application modularization, emulation, and so on. Convergence, on the other hand, tries to bring diverse information assets – databases, mail stores, documents – under unified management. The coming Information Grid unites these opposing drivers.

The drive behind virtualization is the lowering of cost. Today's emerging grid computing environments enable not only the virtualization of IT resources such as storage, bandwidth, CPU cycles (supporting the ad hoc provisioning, on-demand deployment, and decentralized management of the resources), but also allow looser couplings between applications and modules, which are no longer assumed to be monolithic clients and servers. Loosely coupled applications will run different modules on different nodes of a virtualized IT fabric, invoke functionality from remote Web services, exchange self-describing marked up data, and orchestrate the behavior of diverse process modules. XML technologies underpin loosely coupled grid-computing applications. Within the data center, the first generation XML Web services–based service-oriented architectures (SOAs) are already in development.

Convergence, on the other hand, seeks to bring together the management of all of your data assets. Today, less than 10 percent of the world's information is managed, and most of what is found to be valuable to manage – capture, store, index, search, analyze, share, and repurpose – falls into the category of traditional rows and columns such as structured data. Being able to manage the remaining data is what convergence is all about. Here again, XML technologies underpin the renaissance. In XML we finally have a data model that is capable of addressing highly structured data (rows and columns), textual unstructured data (documents), and anything semi-structured in between (messages, template-based business data documents, or metadata). Document-intensive industries are already benefiting from standardizing their document formats on XML. Content-creation vendors are XML-enabling their tools to make it easier to capture information in content repositories. Vendors are XML-enabling business intelligence tools, application servers, enterprise portals, and other infrastructure products to make it easier to share and repurpose XML-based information.

The real driver behind convergence is better business intelligence across all assets. When unstructured information becomes a managed resource, it can be integrated into more day-to-day organizational processes, such as search and compliance, which are really types of business intelligence. Users can search across information that was previously stored in silos, such as file systems, document repositories, Web sites, and e-mail. Collaborative processes can be automated. Compliance policies – privacy, information life-cycle management, and audit – can be implemented uniformly across all organizational assets.

XML's applicability to both virtualization and convergence allows the industry to make progress on both fronts without the need for multiple disruptive paradigm shifts. Moving toward a new data-management architecture based on XML-backed information repositories distributed across XML/SOA fabrics will be a key future step for organizations. This architecture, which combines virtualization and convergence, can be called the Information Grid.

The Information Grid and Its Components

Grid computing can virtualize any IT resource, including infrastructure, applications, and information. In the Information Grid, resources span all of the data in the organization, as well as all of the metadata required to make that data meaningful. This data may be structured, semi-structured, or unstructured; stored in any location, such as databases, local file systems, or e-mail servers; and created by any

AUTHOR BIO

Sandeepan Banerjee is director of product management in Oracle's Server Technologies division. He is responsible for SQL, XML, and Text Search infrastructure, and especially their convergence into one platform for all data. Sandeepan has worked with database technologies for over 15 years, and the majority of them have been with Oracle.

application. The vision for the Information Grid builds on technologies such as semantics, distributed query, and distributed data management. The goal is to enable organizations to view all of their assets in a smooth continuum, from the Internet to the intranet, with uniform, semantically rich access.

Application Grid vs. Information Grid

Within an Application Grid, individual modules run on different parts of the infrastructure, with sharing of application state and control enabled via Web services. Each module, however, may be still tightly coupled to its data – database, file-system, e-mail server – and intelligence about the data has to be compiled into the application module. An Information Grid, in contrast, is self-describing: the application modules can discover what sources exist, what data they possess, what the life cycle of that data is, and how that data should be interpreted. The Information Grid builds on the Infrastructure and Application Grids.

Let's say a manufacturing organization is interested in tracking product defects. The defect reports come into the organization in a variety of ways – customer e-mail, news stories, phone calls to support centers, and so on. At a pure application level, the organization could build e-mail-analysis, RSS-feed-search, or CRM defect-tracking modules to be dispatched across the grid, with each module hardwired to analysis of exactly one kind of data. However, if new kinds of defect reports occur with unpredictable frequency (suddenly Internet blogs become a major source of defect information), then modules that are hard coded to a particular kind of data are proven to be fragile, and the Application Grid is not successful. An Information Grid where the defect reports can describe their own meaning, and modules interact with the defect reports to understand their semantics, appears to be more flexible. The following are the components of the Information Grid.

Infrastructure Provisioning and Failover

What are the major components of the Information Grid? At the very basic level, any grid involves the virtualization of resources. Infrastructure Grid resources include hardware resources such as storage, processors, memory, and networks, as well as software designed to manage this hardware, such as databases, storage management, system management, application servers, and operating systems. Provisioning of infrastructure resources involves pooling the resources together and allocating to the appropriate consumers based on policies. For example, one policy might be to load-balance processing power across a farm of Web servers depending on the amount of processing demanded by each, thus treating the overall processing resource as a single pool and allocating that resource through supply and demand. In addition to the cost savings that accrue from better overall CPU utilization, the spreading of computing capacity among many different computers or spreading storage capacity across multiple disk groups removes single points of failure.

SOA and Business Process Management

SOAs underpin application virtualization in the Information Grid. The foundation of SOA is a set of independent, well-defined encapsulations of software functionality that can be invoked over a network using heterogeneous platforms and execution environments. SOAs connect these independent services toward a larger purpose, where the services must occur in a particular order. SOAs also orchestrate execution in correct sequence; languages such as Business Process Execution Language (BPEL) provide a standard for orchestrating processes into complex business flow.

SOAs are implemented using XML-based Web services standards. Web services are successful where earlier distributed computing architectures have failed for three reasons: simpler standards, broader adoption, and looser coupling. Web services are not only based on simpler standards (HTTP, SOAP) than, say, CORBA, but they have also been broadly incorporated into packaged software and adopted by companies across many industries.

Repository: Metadata, Schema, and Service Management

The Semantic Web community has found out that the missing link to effectively sharing and reusing data on the Web is the lack of machine-readable standards for semantics (i.e., meanings) of Web (typically HTML-based) content. Thus, the Semantic Web is often associated with specific XML-based standards for semantics, such as Resource Description Framework (RDF) and Web Ontology Language (OWL). Within enterprises, almost every product or service is looking to provide an "XML out" that publishes data in a self-describing, standard way for use by other applications. From financial reporting (XBRL) to Web site feeds (RSS) to legal information exchange (LegalXML), XML is the dominant standard for interchange today. In addition to the exchange format standards, management standards such as Access Control Markup Language (XACML) and digital intellectual property rights management (XRML) are also emerging.

"XML's applicability to both virtualization and convergence allows the industry to make progress on both fronts without the need for multiple disruptive paradigm shifts"

The Information Grid also requires semantic information to make each data resource accessible to any process in the Application Grid without requiring any a priori coupling between the data resources and the Application Grid processes. In practice, this relies on metadata describing the meaning of data and relationships among data elements, as well as the implementation of exchange formats and management standards.

The relational database was one early implementation of metadata technology. Unlike its predecessors – the network and hierarchical databases in which all relationships between data had to be predetermined, the relational database enabled flexible yet predictable access to a general-purpose information resource.

XML is the next evolution in the world of metadata. The brain of an Information Grid is an XML Metadata Repository. This repository (which may be physically distributed across nodes and disks) keeps track of the

information. It helps organize all of the resources participating in the Information Grid into hierarchical relationships (the invoice records sitting in database-A logically belong to a folder named Customer sitting on file-system-B, a description of that customer is to be found in CRM-application-C, with the latest interaction recorded in e-mail-server-D, and the connections being automatically deduced from XML tags carried by the data).

The metadata can form different kinds of ontologies. Ontologies specify concepts; ontologies can not only be about a domain (a defect can be an actual failure or a possible failure), but also about tasks (how to compute a possible failure's probability), personalization (different views of a defect, from the legal, support, marketing, or finance perspectives), argumentation (why the defect data was collected, why it was modelled in the way it was, and who agrees to it and who dissents), and so on.

The repository also provides services like event management (what to do if the customer is deleted), business rules (how to determine if the customer qualifies for a volume discount), versioning (issue a new version of the last invoice reflecting the volume discount), access control (who can see the customer's credit card number) and so on. In any distributed system names are used to refer to objects such as computers, services, or data. Typical naming services such as the international X.500 naming scheme or DNS (the Internet's scheme) provide a uniform namespace across the grid. The XML Repository also supports a standard naming service.

The latest generation of relational databases has now evolved to include the XML data model, and several of them support the XML Schema standard for defining exchange formation. Today's best databases, however, also include built-in XML Metadata Repository functionality, thus supporting event management, business rules, versioning, access control, and rights management.

Semantic Crawlers, Search, and Query

On the Web, search engines deploy crawlers or spiders to deduce metadata about HTML pages and index them so that keyword searches can be performed across Web sites. Search servers provide the same opportunity within a grid. On the Information Grid, semantic crawlers extract metadata from the assets as they are crawled (exploiting markup and also employing various heuristics), and the best ones can induce relationships between items through text mining techniques. While crawling across messages in an e-mail server, a semantic crawler might deduce that the presence of the word complaint, refund, or other such term in an e-mail message indicates an unhappy customer. Later, when a query is initiated for a customer name through a keyword search interface, the search server can color-code the search results, thereby indicating how "happy" the customer is. This is the business intelligence value of the Information Grid showing through.

> "XML is the next evolution in the world of metadata"

Currently, search engines are poor examples of semantic processing – typically two different users with the same query will get the same result, even if one was searching for an insect (cricket) and the other for a game (cricket). Humans can generally understand which hit is about what, but automations built around search-hit-lists fail due to the high semantic ambiguity. Ideally, the search query will be qualified by the user's context, the data described by the creator's context, and the two matched to give unambiguous results.

Within an intranet, the crawlers also need to be able to respect and enforce security, information lifecycle management (ILM), and privacy policies. The best search servers today combine security, the semantic relevance of returned results, and the ability to intelligently present as much contextual information that exists.

As information is distributed around the Information Grid, the other consideration is integration. What if, in addition to keyword search style queries, one also wants to perform "joins" across different data sources on the grid, or perform advanced "slice and dice" queries a la SQL or OLAP? The assets in an Information Grid are not all relational, so simply using a distributed SQL engine is not feasible; however, since the assets are all expressible as XML, a query language based on XML can be used to join across them. The emerging XQuery standard, combined with connectivity technology such as the J2EE Connector Architecture (JCA), can be used to query across joins of, say, documents (expressed as XML) and relational data (also expressed as XML). The strength of XQuery is that by using the XML data model intelligently it can express queries across all these kinds of data, whether physically stored in XML or viewed as XML via the Information Grid. You can now ask to join customer name from an invoice database to an e-mail system to see how many people have sent in questions on their invoices, and so on.

Information retrieval also goes beyond keywords. A picture is worth a thousand hit lists, and visualization techniques in the Information Grid are providing pictorial summaries of queries: maps that are color coded to show relative incidences of results, charts that indicate the distribution of data for a query, theme-viewers that highlight the most important concepts in an e-mail or document. It is a brave new world.

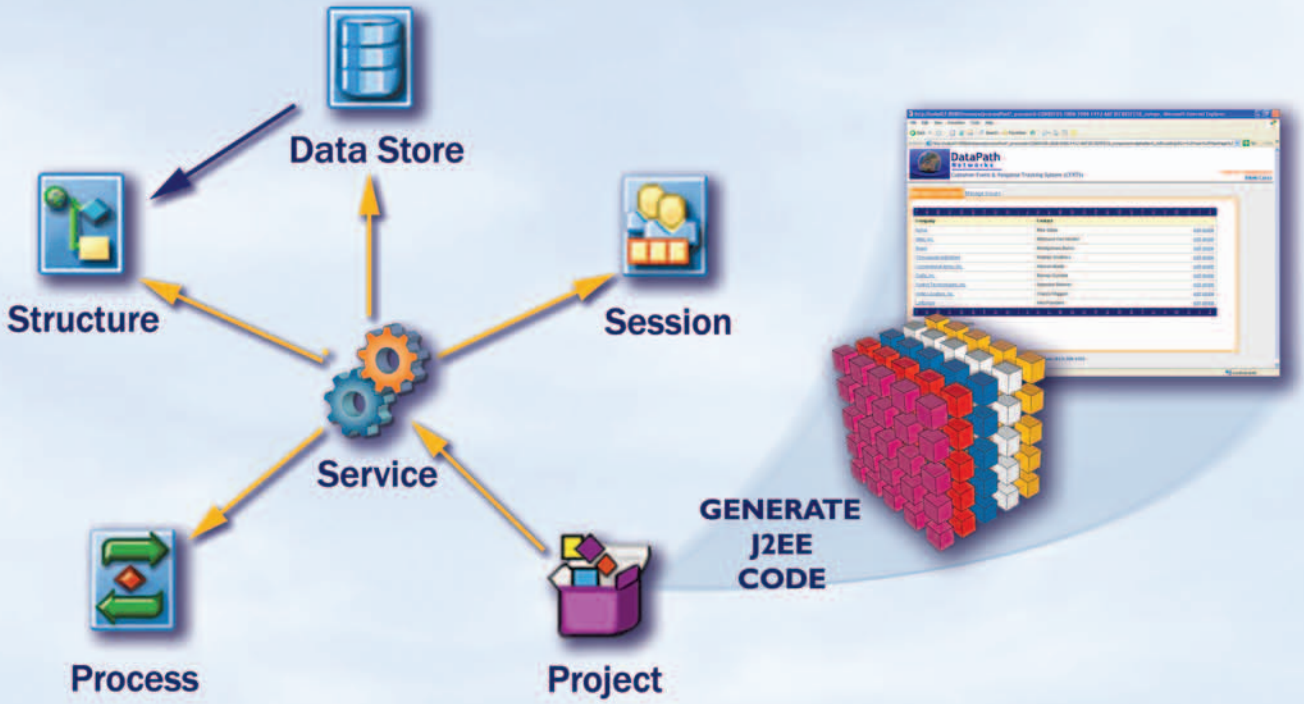
Where Do We Go from Here?

The Information Grid combines virtualization and convergence for holistic data management to extract full potential from all enterprise information sources and easily exploit new information relationships as conditions change. The benefits are not only the cost savings and reliability that accrue through the virtualization of infrastructure and application resources, but functionally as well: rich, end-to-end business intelligence across all information assets. Organizations must plan beyond the virtualization of CPU or storage, and look toward the coming XML-ization of all data and metadata across Information Grids.

sandeepan.banerjee@oracle.com

WANTED: SOA
NO EXPERIENCE NECESSARY

BUILD, GOVERN, & *DEPLOY*
Business Level Solutions, Processes, and Services



CALL NOW
For Case Study Details on How to Implement SOA Quickly and Easily

"From a support and knowledge standpoint, Skyway is untouchable. What's immeasurably beneficial is how quickly Skyway responds; it is truly phenomenal. A lot of people only say they can do what Skyway delivers. With Skyway you will do more with less." — Jim Garcia, CIO, Enporion

"Developing with Skyway Software gives us a much more rapid response time to our business community, it has a lower TCO, and it accelerates SOA and Web Service adoption throughout BAT."
— Kevin Poulter, British American Tobacco, a $30 billion powerhouse

White Paper — Case Study — FREE Evaluation Download

www.skywaysoftware.com

Skyway software

(813) 288-9355

Never mind what your mother told you. In your world of multiple platforms and databases, it's not only *okay* to talk to strangers, it's *essential*. That's where z/XML-Host™ comes in. z/XML-Host makes it Easy for your mainframe 3270 applications to talk with any other platform using XML and SOAP-based Web Services. This includes .NET and Java applications--even Microsoft Excel can access your mainframe data Easily. And since z/XML-Host runs directly on your mainframe, you can meet strangers on the same day you complete the Easy installation.

Visit **www.illustro.com/strangers.htm** and learn more. Even download a fully-functional version of the software to try for Free.

Internet-enabling your mainframe with illustro's z/XML-Host? **Now you're talking!**

Works with .net Java Microsoft Office

illustro™ SYSTEMS INTERNATIONAL, LLC

See The Light.™

www.illustro.com • Toll-free (866) 4-illustro or (214) 800-8900