SYS-CON

NOW INCORPORATING XML JOURNAL

WebServices JOURNAL

.NET J2EE XML

April 2005 Volume 5 Issue 4

Supporting the Business Process Lifecycle Using Standards-Based Tools

Closed-loop BPM solutions

pg.10

Web Services Strategy – SAP Platform

It's not your father's SAP

pg.24

Enterprise Semantics: Aligning Service-Oriented Architecture with the Business

Bringing agility, transparency, and integration to your Web services infrastructure

pg.30

REST vs. SOAP: The Battle of the Web Service TITANS

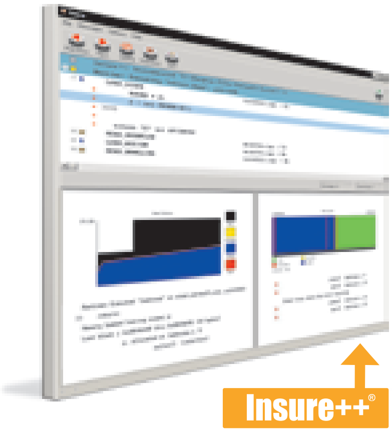
Picking a winning Web service approach

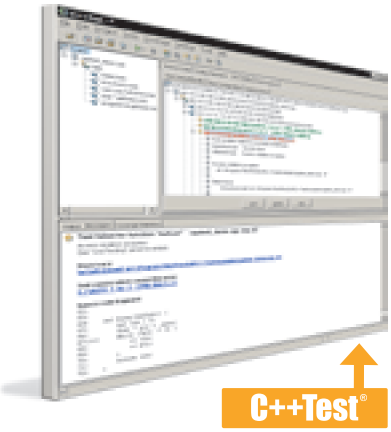
pg.40

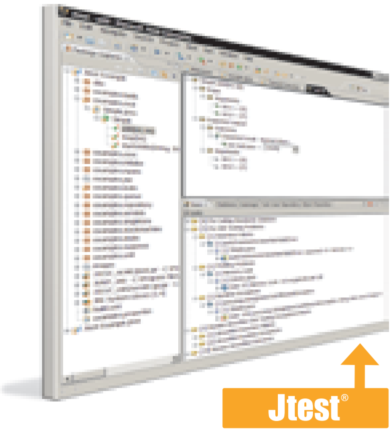
Announcing WEB SERVICES EDGE Fall Conference Series

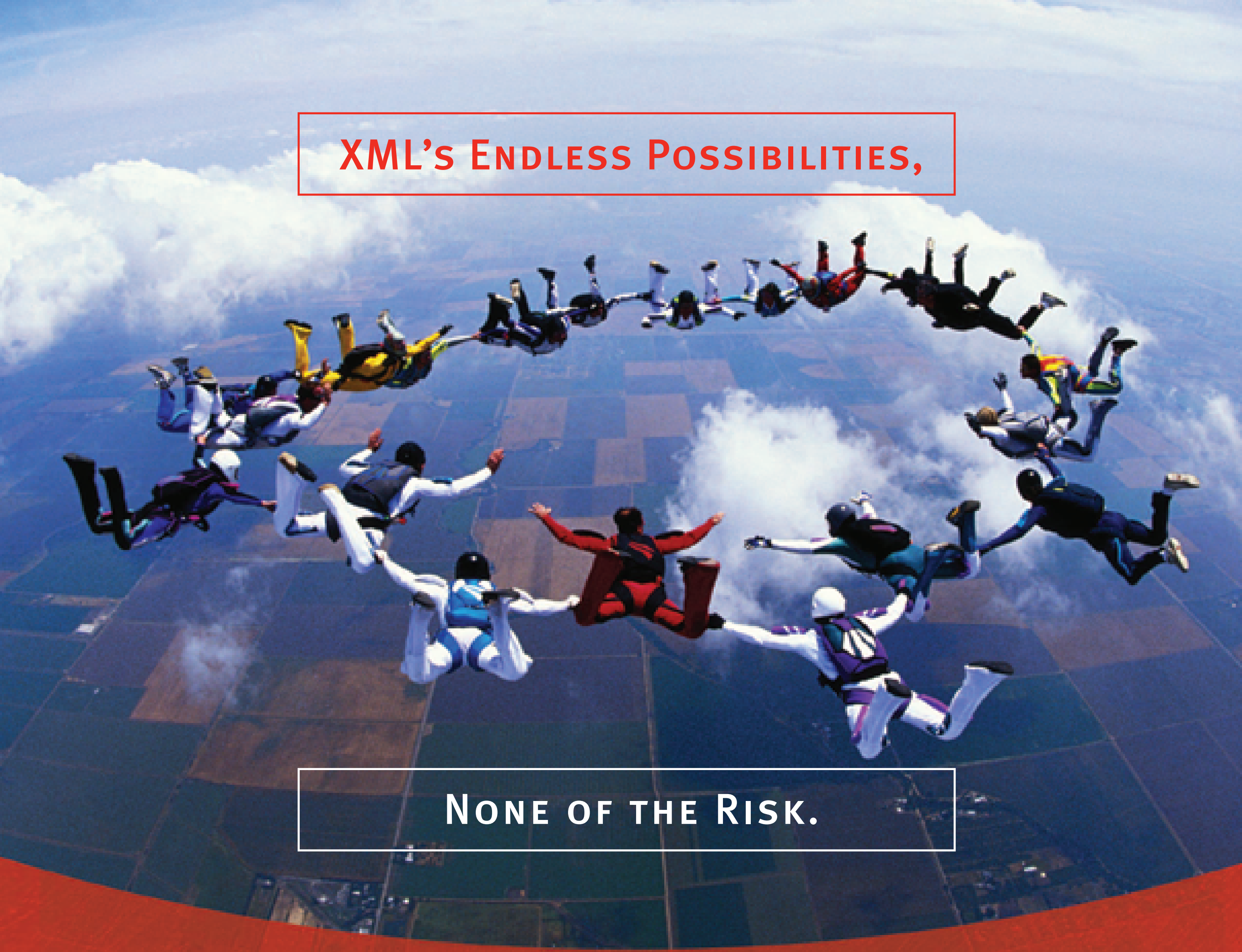
pg. 53

THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES • WSJ2.COM


Insure++®


C++Test®


Jtest®

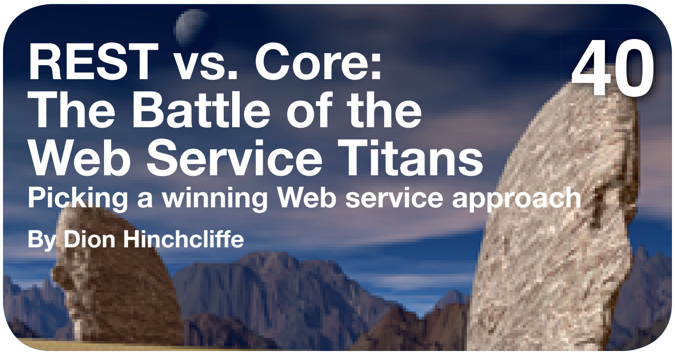

XML's Endless Possibilities,
None of the Risk.

WebServices Journal
.NET J2EE XML

April 2005 Volume 5 Issue 4

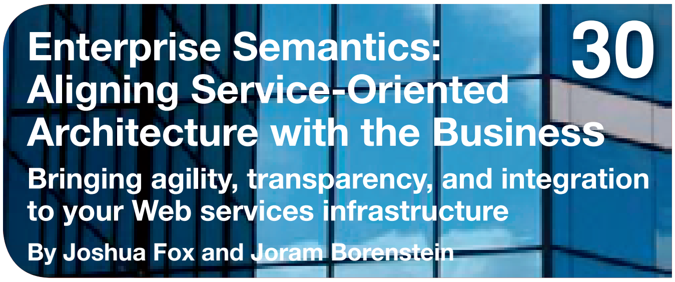
Inside WSJ

XML JOURNAL


REST vs. Core: The Battle of the Web Service Titans 40
Picking a winning Web service approach
By Dion Hinchcliffe

Supporting the Business Process Lifecycle Using Standards-Based Tools 10
Closed-loop BPM solutions
By Bhagat Nainani, Mohamad Afshar, and Jog Raj

Web Services Strategy – SAP Platform 24
It's not your father's SAP
By Scott Campbell


Enterprise Semantics: Aligning Service-Oriented Architecture with the Business 30
Bringing agility, transparency, and integration to your Web services infrastructure
By Joshua Fox and Joram Borenstein

From the Editor
English or Metric?
Sean Rhody 7

Industry Commentary
SOA, Web Services, and GDM
Ajit Sagar 8

WSJ: Product Review
Skyway Builder from Skyway Software
Reaching for the sky
Brian Barbash 18

WSJ: WS Standards
The State of Standards
Do they help us do our job, or do they just get in the way?
Anne Thomas Manes...................20

WSJ: WS Process
Adeptia BPM Server 4.1
More than Enterprise Application Integration on steroids
Paul T. Maurer..............................36

WSJ: WS Core
Understanding Coupling in the Context of SOA
Promoting protocol-based integration
David Linthicum...........................44

XML JOURNAL SUPPLEMENT

This Month

What Lies Beneath

Data(base) considerations for service-oriented architectures

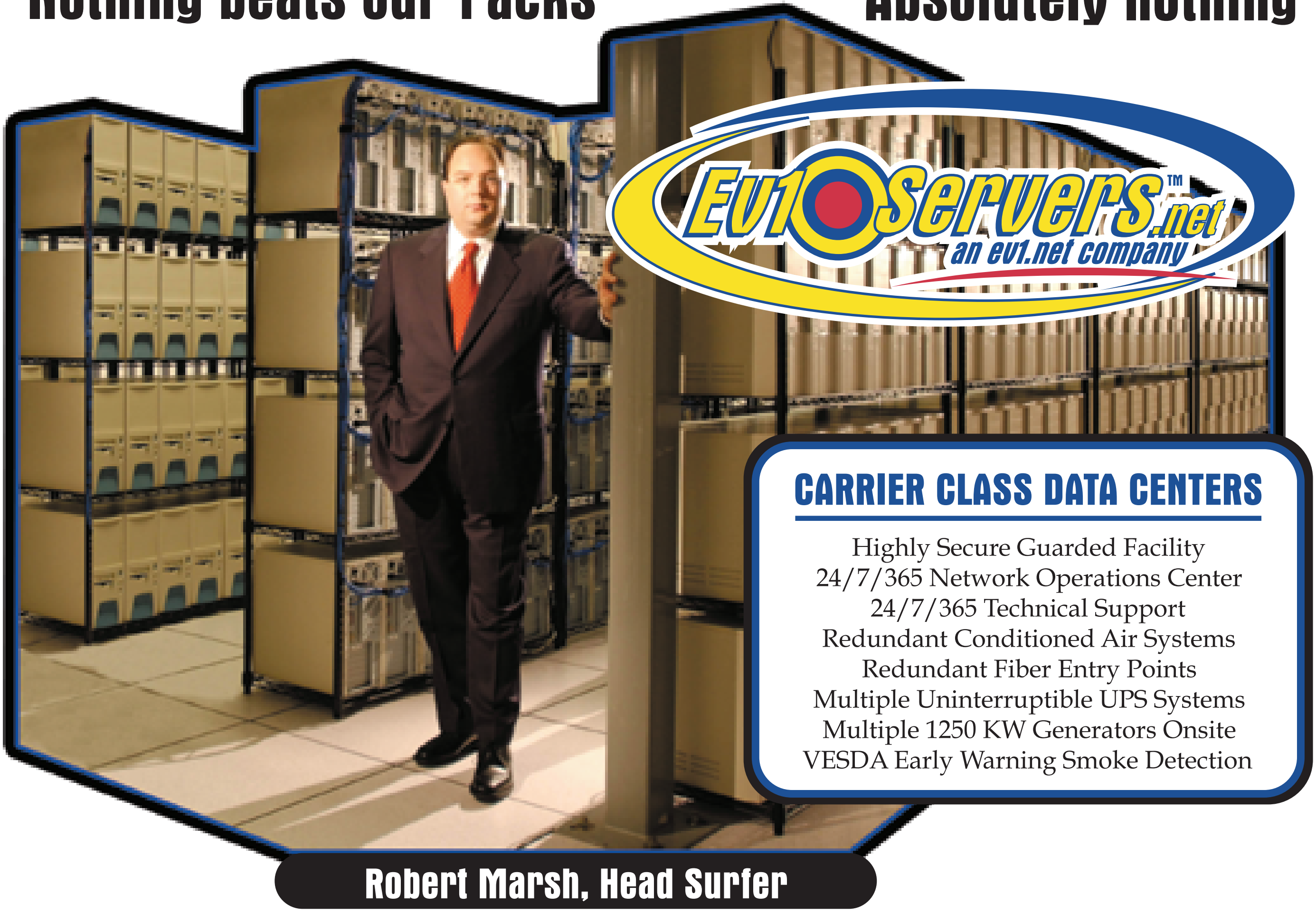
XML-Based Interop, Close up

What Lies Beneath
RAJAN CHANDRAS
48

XML Schema May Not Always Be the Right Tool for the Job
MICHAEL WACEY
52

Overview and Analysis of the Department of Defense Discovery Metadata Specification (DDMS)
MICHAEL SICK
54

Middleware is Everywhere.
Can you see it?
1
4
5
2
3



WebServices JOURNAL
.NET J2EE XML

EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
XML EDITOR
Hitesh Seth
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarbash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
RESEARCH EDITOR
Bahadir Karuv, Ph.D Bahadir@sys-con.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
Michael Wacey mwacey@csc.com

INTERNATIONAL TECHNICAL EDITOR
Ajit Sagar ajitsagar@sys-con.com

MANAGING EDITOR
Seta Papazian seta@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com

ONLINE EDITOR
Martin Wezdecki martin@sys-con.com

PRODUCTION
PRODUCTION CONSULTANT
Jim Morgan jim@sys-con.com
LEAD DESIGNER
Richard Silverberg richards@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTORS
Andrea Boden andrea@sys-con.com
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Mohamad Afshar, Brian Barbash, Joram Borenstein,
Scott Campbell, Rajan Chandras, Joshua Fox, Dion Hinchcliffe,
David Linthicum, Anne Thomas Manes, Paul Maurer,
Bhagat Nainani, Jog Raj, Sean Rhody, Ajit Sagar,
Michael Sick, Michael Wacey

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

SYS-CON MEDIA

English or Metric?

WRITTEN BY
SEAN RHODY

Despite the fact that I have a toolbox full of tools, I'm not a mechanic. I have wrenches and sockets in all sizes and shapes, and because I'm not inherently tidy, they tend to get all mixed up in the toolbox as I use them. So as you may expect, when it comes time to tighten a bolt, I have to go rummaging around in the toolbox, trying to find the right socket to use.

What makes matters more painful for me is the fact that sometime in the past two different standards of measurement were implemented for sockets. There is English measurement, which is based on increments of the Inch, and Metric, which naturally is based on the Meter. The problem is that there is usually no indicator on the bolt as to what size it is, or which unit of measure, and therefore system of sockets, is to be used. Invariably, I end up nearly stripping the bolt head before I realize that it's the wrong unit and finally find the right socket to use.

Sometimes the only thing worse than no standard is more than one.

We're back to the issue of standards within *WSJ*. Once again we will examine the state of Web services as they relate to the standards, which have been proposed, ratified, and implemented.

To start with, we can be happy that the basic standards of Web services are alive, well, and relatively clear of competitors. If all you need is basic connectivity, you can easily do meaningful work using just SOAP, XML, WSDL, and UDDI. UDDI Version 3.0 has recently been ratified, adding some significant features to the basic Web services registry, such as support for public and private registries. At the same time, some of the features of UDDI overlap with WSDL, and it's still not clear that UDDI is absolutely necessary for anything. Searching for a service may be technically interesting, but in most cases it's not part of the actual business implementation. Within organizations services move rarely, if at all, so the need to dynamically locate a service is usually not an imperative business requirement. The overlap is like having two sets of sockets, some of which are nearly the same sizes and in a pinch could be made to serve. Still, it's easier to just use the right one.

Even SOAP is not immune to competition, although it seems to be the predominant standard. REST is another standard that purports to be better than SOAP, depending upon which pundit you read. Again they do much of the same, in slightly different ways, making it harder and harder to decide on the standards of even the basics of Web services.

The challenge that all of these competing, nearly equivalent standards present is that they in fact reduce a standard to something less concrete. Where you could expect a single standard to be the only way the overwhelming majority of people do things (there is always someone who just won't go the same way as everyone else), when multiple standards exist, that uniformity that adds value for this particular set of software is diluted.

The biggest benefit of Web services is that they are implemented by almost every platform and software vendor. Early on the WS-I was formed when it was recognized that even with the single standards that were out, not every implementation really got it right. That's why the WS-I is so valuable; it makes interoperability more likely.

It's also why every new competing standard reduces the usefulness of the standard with which it competes.

My toolbox needs organizing – I'm a software architect, not a mechanic or carpenter. My real working tools are software and standards. While I can live with my toolbox at home being a little unorganized, I can't say the same about my working tools. I need them to be clear, simple, easy to use, and unambiguous. And so do you.

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
Kristin Kuhnle kristin@sys-con.com
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
Monique Floyd monique@sys-con.com

sys-con.com
CONSULTANT, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Matthew Pollotta matthew@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com
ACCOUNTS RECEIVABLE
Steve Michelin smichelin@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 Publications, Inc., reserves the right to revise,
republish and authorize its readers to use the articles
submitted for publication.

SYS-CON MEDIA

SOA, Web Services, and GDM

WRITTEN BY
AJIT SAGAR

A couple of months ago I got an e-mail inviting me to keynote an SOA/Web Services conference in Beijing. My immediate reaction was – "Good. China has reached the stage where it's hosting international conferences on the subject." Actually, 2005 marks the fourth time this particular conference is being held. While China is growing rapidly in IT, it's fairly new to the services game. My subsequent thought was – "This is great, the pitch for service orientation is becoming a reality."

I say this because although SOA is rapidly becoming reality, and Web Services are proving to be the ideal platform to host SOA, SOA's true potential will only be realized when the service itself is distributed geographically, and can be realized not only by architecting solutions that are technology-agnostic, but also by implementing geographically distributed cross-cultural solutions, thus truly uniting the distributed enterprise. If it smells like utopia, that's because it is; we can but aspire.

The prospect of going to China and talking about SOA and Web Services was very exciting, so I accepted the invitation. Now came the hard part – what to talk about? I could talk about WS standards, security, governance, and strategy – there's no shortage of topics to present. But somehow none of these really hit home. I wanted to talk about a subject and offer a unique perspective on something I had a deeper perspective on. Then it suddenly struck me. I'm going to China, a country ripe for a revolution that will support SOA initiatives around the world.

Most of the initiatives in place today target the development of services and their deployment in an organization. The actual development of these services lends itself to outsourcing – primarily because of economies of scale, the resource crunch, and skill set availability – and all these lead to the main pain point of organizations – the cost of operation. SOA is causing a major change in the development of services and is inherently promoting outsourcing. The current outsourcing model has a mechanism – the Global Delivery Model (GDM). GDM is basically working in orthogonal directions. Established system integrators based in the US such as IBM and Accenture are offshoring development, and companies that started at the delivery end of the spectrum, such as Infosys, TCS, and Wipro, are moving towards a higher level of strategic engagement. But the model is the same – offshoring development of services that are deployed here in the US.

At Infosys, we are evolving the GDM that we put in place for traditional projects such as platform migration, development, and maintenance, and are applying the same model to developing SOA strategy and blueprints for our customers. In other words, GDM is being leveraged at the strategic, not just the tactical, level. It can be called a technology competency offering. The skill sets for these distributed groups are in architecture, analysis, IT strategy, and mainstream development. Of course, for an end-to-end offering, development has to support the downstream strategy.

However, all this is still focused on developing at an offshore location and deploying at the client. The next revolution will be distributing the hosting of these services in a shared services model. The infrastructure to support this in terms of people, process, and management is there. The business model is not.

Coming back to China, the economies of scale will let China replicate the GDM model and offer similar services at a global level. All the pieces are there. They just need to be connected.

So I'm en route to China as I write this. On to SOA, Web Services, and the Global Delivery Model!

Resources

- *For details about Web Service & SOA China 2005:* www.webserviceschina.com.
- *For details of the session Web Services, SOA, and the Global Delivery Model, please go to:* www.ajitsagar.javadevelopersjournal.com/read/1088114.htm.

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. He has been a programmer, lead architect, director of engineering, and product manager for companies from 15 to 25,000 people in size. He has served as JDJ's J2EE editor, was the founding editor of XML Journal, and has been a frequent speaker at SYS-CON's Web Services Edge series of conferences. He has published more than 75 articles.

■ ■ ■ ajitsagar@sys-con.com

Is your SOA really

Building, testing and maintaining SOA solutions requires a flexible and collaborative approach to diagnostics. Finger pointing and confusion occur when all parties do not have a complete and common understanding of a problem. Mindreef® SOAPscope® connects developers, testers, support personnel, and operations by combining the testing and diagnostic tools for each discipline and making it easy to share complete problem data. SOAPscope is the only end-to-end diagnostic system for Web services.

Developers • Testers • Operations • Support

Mindreef SOAPscope

Test Agile testing approach for both developers and testers allows what-if scenarios without coding or scripting. Includes the industry's most complete interoperability suite and the flexibility to really test the edge cases.

Capture Easily capture and assemble complete problem data into SOAPscope's workspace. Capture SOAP messages off the wire, test cases, or reproduced problems. Capture a snapshot in time of a WSDL including all related resources.

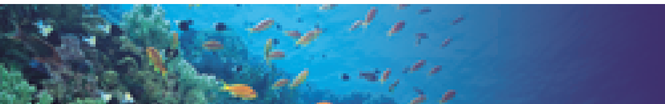
Share Exported workspaces support sharing among team members preserving complete problem data including messages, WSDLs, and notes. Sharing workspaces is a great way to support web service users.

Solve Diagnostic suite of analysis, comparison and visibility tools simplifies the understanding of complex XML, yet are powerful enough for the industry's most protocol-savvy experts.

Try SOAPscope **FREE** at www.Mindreef.com/tryout

"SOAPscope has been invaluable in testing and diagnosing our enterprise scale Web Services. The new Workspace feature in SOAPscope is an excellent way to support customers and collaborate as a team. A must have for every WS-Developer"

Simon Fell
Senior Member of
Technical Staff at
salesforce.com and
author of PocketSOAP



Mindreef, Inc.
www.Mindreef.com

Supporting the Business Process Lifecycle Using Standards-Based Tools

Closed-loop BPM solutions

Business processes integrate systems, partners, and people to achieve key strategic and operations objectives. Examples of business processes include getting and filling orders, processing invoices, reconciling shipping notices and received goods and processing insurance claims and loan applications. The Holy Grail of enterprise computing is adaptive business processes that can be defined, refined, and optimized to respond to changing business environments, government regulations and competitive pressures. This vision has followed us through the evolution of mainframes, Management Information Systems (MIS), packaged applications, J2EE-based application platforms, business process management systems (BPMS) and now, Service Oriented Architectures (SOA). We are getting there!

WRITTEN BY
BY BHAGAT NAINANI, MOHAMAD AFSHAR, & JOG RAJ

Many solutions exist for designing and deploying business processes. Some are derived from proprietary BPMSs or workflow engines; others have been built as process execution engines for Web Services and SOA, and are usually based on the Business Process Execution Language (BPEL) for Web Services. All of these solutions have a common goal: to support the business process lifecycle from modeling and design to implementation and monitoring, and then back to design. Full lifecycle support enables continuous process improvement. Being able to iterate continually through the process lifecycle is called closed loop. Systems that support this are called closed-loop business process management or closed-loop BPM solutions.

Large enterprise software vendors offer closed-loop BPM solutions, while niche vendors offer point solutions that tackle part of the cycle, such as modeling. But can you put together best-of-breed tools from multiple software vendors without spending a lot of time integrating them? In other words, are there any interoperable standards-based tools that support the business process lifecycle? The good news is that with standards such as BPEL and Business Process Modeling Notation (BPMN) you can implement standards-based, closed-loop BPM.

BPEL – a standard, portable language for orchestrating services into end-to-end business processes – builds on a decade of progress in business process management, workflow, and integration technologies. It's built from the ground up around XML and Web Services and is supported on both the Microsoft .NET and Java platforms. BPMN, as an amalgamation of best practices in the business process modeling community, provides a simple, standard means of communicating process information to key stakeholders.

THE BUSINESS PROCESS LIFECYCLE

The business process lifecycle consists of the following steps:

- ***Model and Simulate:*** Business process owners create a high-level design of the tasks to be done and a list of the required resources. This is usually done graphically using a drawing package such as Microsoft Visio or a proper modeling tool such as Popkin's System Architect. The modeling tool may also be used to perform optional simulation steps during which hypothetical scenarios are run to identify critical paths and bottlenecks.
- ***Implement and Deploy/Execute:*** Developers convert the business process definitions into an executable process model linking systems, APIs, and people through workflows. The resulting executable process is then deployed to a BPEL or BPM engine for execution.
- ***Monitor and Optimize:*** Deployed business processes are monitored to measure key performance indicators and other metrics. Process throughput and utilization metrics can be fed into a simulation tool to derive the optimal execution mode by using real data (e.g. historical).

This cycle is shown in Figure 1.

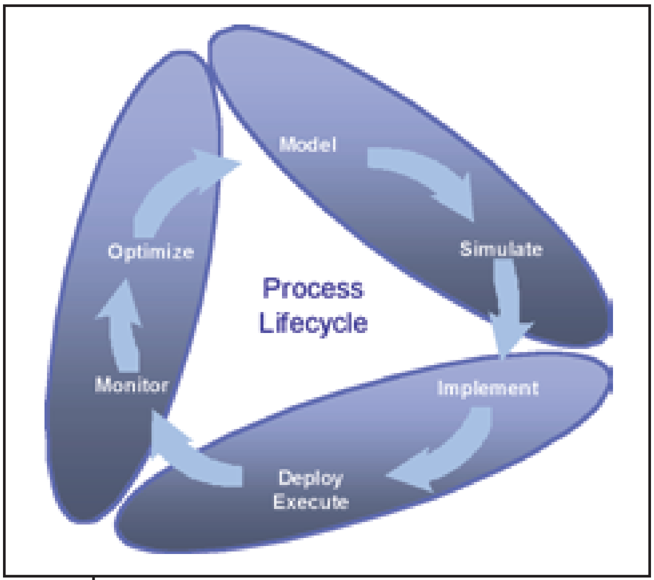


FIGURE 1 Business process lifecycle

Designing and Deploying Processes

There are many ways to implement business processes (including using packaged applications that, for the sake of brevity, we won't discuss here). Figure 2 shows the spectrum of possible approaches for process design and deployment as well as the level of automation and flexibility offered by each approach. The obvious Procedural Code-based Solution (far left) uses SOA and Web Services but uses a development tool to write procedural code. The ideal 100% Standards-based Solution (far right) is a flexible, adaptable solution. Other approaches fall in between.

Obvious Solution: Procedural Code-Based Approach

How do you model, implement, and deploy business processes using a procedural approach? The choices are reasonably clear. The business analyst models the system using a modeling tool (for example, in UML), and the programmer incorporates existing services and develops new ones with J2EE. Although the analyst creates a high-level process model, it doesn't map directly to an executable process model, so the programmer has to implement the processes based on his understanding of the process model. Processes built in this way are hard to develop and error-prone because of the disconnect between the high level model and the completed process.

Further, systems built this way are difficult to change because you have to rewrite code. It's also hard to get performance metrics at the business level, such as the length of time it takes to ship an order – but these metrics are important to business managers. Without them, they are left "blind" to the real state of the business. Besides, changing business policies isn't easy since policies are hard-coded in application logic, making automated change impossible.

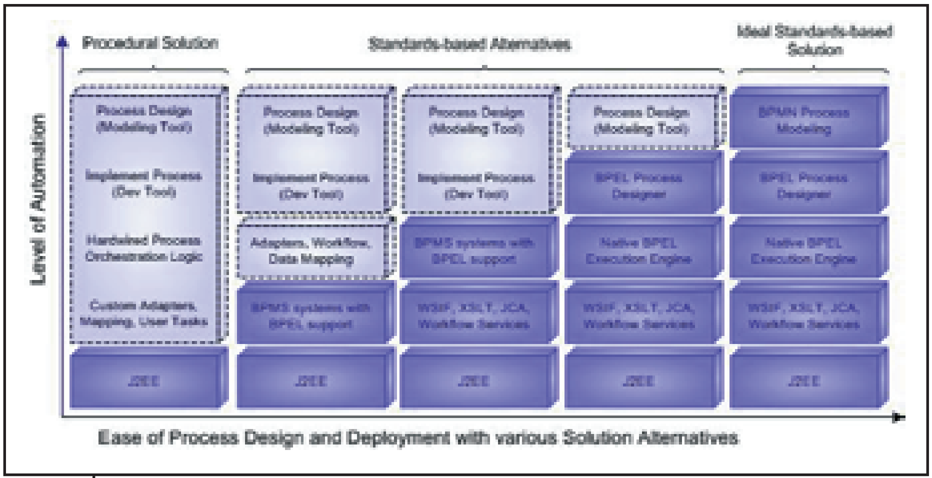

Procedural Solution
Standards-based Alternatives
Ideal Standards-based Solution
Level of Automation
Process Design (Modeling Tool)
Implement Process (Dev Tool)
Hardwired Process Orchestration Logic
Custom Adapters, Mapping, User Tasks
J2EE
Process Design (Modeling Tool)
Implement Process (Dev Tool)
Adapters, Workflow, Data Mapping
BPMS systems with BPEL support
J2EE
Process Design (Modeling Tool)
Implement Process (Dev Tool)
BPMS systems with BPEL support
WSIF, XSLT, JCA, Workflow Services
J2EE
Process Design (Modeling Tool)
BPEL Process Designer
Native BPEL Execution Engine
WSIF, XSLT, JCA, Workflow Services
J2EE
BPMN Process Modeling
BPEL Process Designer
Native BPEL Execution Engine
WSIF, XSLT, JCA, Workflow Services
J2EE
Ease of Process Design and Deployment with various Solution Alternatives

FIGURE 2 Process analysis, modeling, deployment, and execution options

Ideal Solution: Process Modeling using BPMN, Implementation using BPEL

So what does the ideal solution for modeling, implementing, and executing a business process look like? The ideal solution models business processes using BPMN and implements and executes them using BPEL. A business analyst can use a business-process modeling tool (such as Popkin's System Architect) and then export the process model as BPEL. This can then be imported into a BPEL-compliant SOA/process development tool (such as BPEL Designer in Oracle JDeveloper 10g) to define and implement the process further.

BPMN, developed by the Business Process Modeling Initiative (BPMI.org), provides a notation that all business users can easily use and understand. These users include business analysts who model business processes, technical developers who build systems that implement those processes, and managers who must understand and review process diagrams.

Some key BPMN features include:

- ***Pools and lanes:*** These entities are used to demarcate processes, participants, and systems. They help build multi-level process diagrams.
- ***Events and activities:*** Events are used to represent something that happens in the course of the business; they usually have a cause and effect. Activities are work that's done – either a single task or a set of tasks (sub-process).
- ***Sequence flows and message flows:*** These indicate the order in which activities are done and identify messages exchanged between entities. Associations are used to attach text and other artifacts to flow objects.
- ***Model message and control:*** These data objects model how documents, data, and other objects are used and updated during a process flow.

BPMN objects have a rich set of attributes that enables easy mapping to BPEL. The skeletal BPEL process output from the modeling tool generally consists of process scopes, invoke/receive activities, and partner links to the appropriate services. Further work is required by the programmer, including binding to heterogeneous systems, supporting synchronous and asynchronous message exchange patterns, transforming data and schema, coordinating activity flow,

Use Simulation When Optimizing Business Processes

Let's talk briefly about process simulation. Project teams often simulate the modeled process—running hypothetical scenarios or historical information through the model—so they can gauge overall performance and identify possible bottlenecks. Before a team begins this process, which occurs between the BPMN model creation and BPEL implementation, it must assess the amount of time required to execute each activity, the resources required for each task, and the probability that various events or conditions will occur in the flow.

Simulations typically:

- Provide average elapsed time per transaction, end-to-end throughput, and standard deviation to determine if these are within acceptable Service Level Agreements (SLAs). Think about a telecommunications line provisioning process in which the telco guarantees the customer a 48-hour setup. Simulations can give practitioners and developers greater confidence that implemented processes will be able to fulfill such SLA.
- Identify bottlenecks in the process and resource utilization. For example, there could be bottlenecks in processes that involve human workflow interactions, such as managerial approvals.
- Determine the number of employees and employee hours required to complete tasks to reach specified SLAs.
- Calculate expected failure rates and Six Sigma service-level ratings.

managing exceptions, handling non-deterministic events, defining compensating transactions, and managing version control.

BPEL provides a rich yet simple abstraction for addressing these requirements by adding implementation details on top of the BPMN model.

Here are the steps for turning a high-level BPMN process model into an automated process that can then be deployed:

- Identify Web service operations that are invoked on various services
- Specify XSD types for the messages that are exchanged between different entities
- Model transformation maps to convert data from one representation to another
- Specify endpoint locations and connection parameters for the services involved
- Add fault-handling and compensation logic

Proprietary BPM, workflow, and process integration products can import BPEL. But, a native BPEL execution engine has many advantages over non-native implementations. For example, it won't lose information during the import/export process, and you can switch between model and BPEL code in real-time during development. BPMN and BPEL tools simplify communication between business analysts and developers, enabling them to collaborate more effectively. These tools do this by providing a seamless visual modeling experience from process design through implementation.

Monitoring and Optimizing Processes

Once business processes have been deployed, it's critical to monitor their performance by measuring key process metrics and having real-time visibility into process execution. Monitoring can be done at the operational or business level. Standards are still evolving in the monitoring and optimization domain; however, interoperable platform-based solutions are available. Figure 3 shows the various alternatives for process monitoring and optimization and the level of automation offered by each one.

Operational monitoring generally involves getting details such as process status, messages exchanged, performance, and audit trails. This is typically done via process monitoring metrics presented through a console. Many BPEL engines, process orchestration engines, BPMSs, and workflow engines that support BPEL provide a console for drilling down into the execution of business processes.

Available metrics could include how long it takes to ship an order, how long it takes to get goods from suppliers and pay for them, or how long it takes to provision a DSL line. For example, Oracle BPEL Process Manager includes a sensor framework that can publish process transitions and events to a queue (e.g., JMS) and database tables, which can then be received by a BAM tool. BAM gives you visibility into key business indicators along with user alerts and dashboards to respond in real-time to events or exceptions.

BAM tools can:

- Capture system, application, or external events throughout an organization, and then filter, aggregate, and correlate them to answer key questions about how the business is doing. Events might be customer updates, inventory changes, or purchase orders, and may come from a variety of IT systems.
- Generate metrics, KPIs, and trends. These can be used to identify causes and effects between different parts of the business, or to determine predicted outcomes so that corrective action can be taken more quickly.
- Provide a dashboard with a rich set of real-time visualization features that can display the latest event data and let you drill down for root cause analysis and even take corrective action.
- Send alerts to users, via various delivery channels, based on threshold violations or risks for certain key metrics, as mentioned above. Besides simple notifications, alerts can also be used to trigger automated responses or invoke structured workflows to deal with more complex situations. For example, if DSL line provisioning takes more than 20% of the average completion time in a 48-hour period, this could be flagged to an operations manager, or a set of business processes could be automatically fired directly from the BAM solution. BAM provides the missing link between process execution and redesign - with BAM, you can, for instance, actually find out if a supplier fulfills orders in stipulated SLAs. BAM also

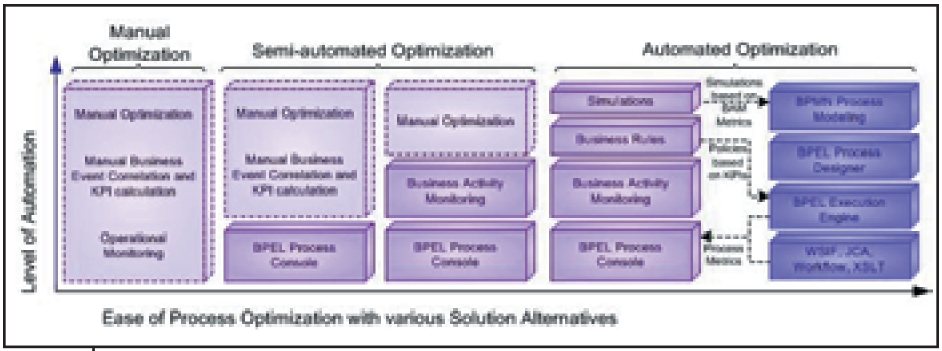


FIGURE 3 **Monitoring and optimization options**

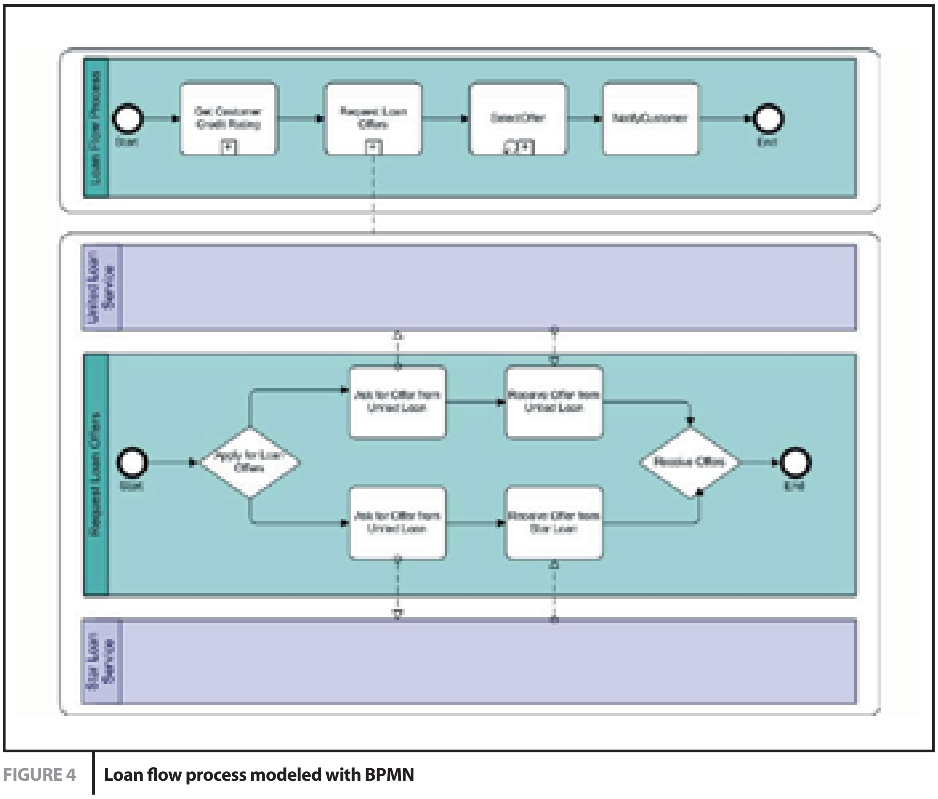


FIGURE 4 **Loan flow process modeled with BPMN**

helps with simulations. Traditionally, process simulations were based on guesstimates, making the results only as reliable as the assumptions made during the modeling phase. With BAM, real-time as well as historical data can be used to model the as-is process and create the optimal to-be process. With successive iterations, the process can be refined to generate significant resource and cost savings. This feedback – closing the loop – is key to achieving business optimization. Feedback can be manual (a person sees trends and takes action based on them) or automated (the BAM system fires events that make in-flight adjustments to processes and systems).

Case Study: Online Loan Application Processing

Consider the following example of a loan flow process deployed at a typical loan broker. The broker accepts a request from a client, does a credit check with an external service, and routes the application to two loan agencies. After getting two offers, the better offer is selected and the customer is notified. In this section, we'll show how you can automate this business process with available solutions based on SOA and Web Service principles using BPMN and BPEL.

In the modeling phase, the business analyst specifies the participants (LoanBroker, CreditRatingService, StarLoanService, UnitedLoanService, and the customer). The loan flow process orchestrates interactions between these services into end-to-end flows. To enable this, the analyst specifies the sequence of events and message flows between these entities using BPMN within the modeling tool, such as Popkin's System Architect. Figure 4 illustrates the high-level BPMN process model and a drill-down into a subset of this process, where a request for a loan offer is routed to the two loan agencies.

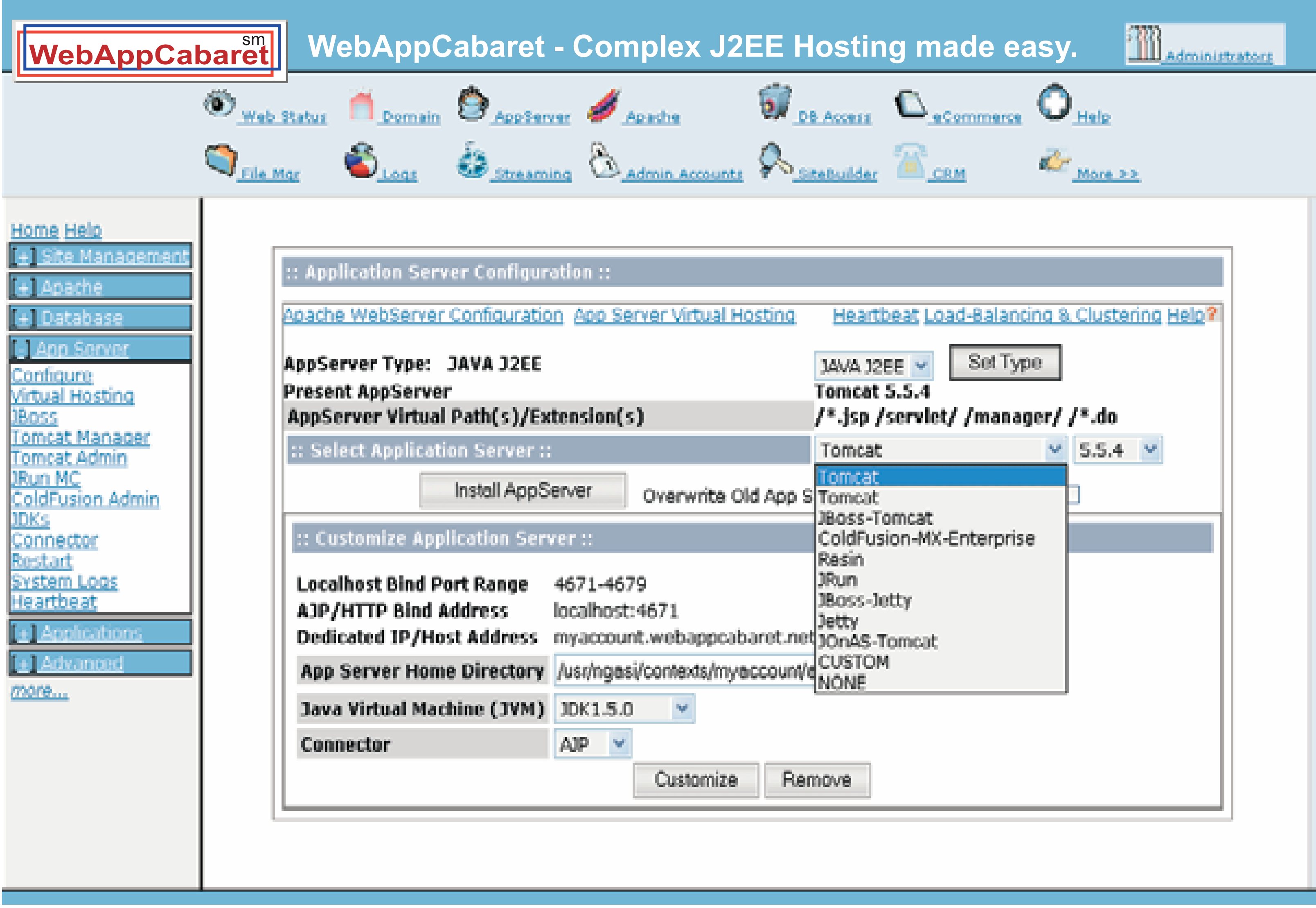
To further analyze this process, you could model the loan application inflow rate and processing time for each loan agency and then run simulations to come up with the throughput or response time for the end-to-end flow. If we also assume that a certain number of loan agents at the two agencies are needed to complete the task, and we assume an average time to process a loan, we can use simulations to help determine resource usage and the number of resources required based on the expected load. Some modeling tools let you run simulations based on the process model. For example, if you're using Popkin's System Architect to model business processes, you can use Popkin's SA Simulator II to run the simulations (i.e., run through the hypothetical scenarios).

After modeling the process in BPMN, you export to BPEL. You can then complete the BPEL skeleton generated from the BPMN modeling tool to include URLs for the services, and XSDs for the loan application and loan offer, and data transformations for the documents exchanged between the services.

Oracle BPEL Designer provides a graphical way to build the loan-flow BPEL process. Because it uses BPEL as its native format, processes built with Oracle BPEL Designer are 100% portable. It's also available as a plug-in for Oracle's JDeveloper IDE, or the Open Source Eclipse IDE, so developers can easily program the code for implementing new services and user interfaces in the same tool as they implement the BPEL processes. Figure 5 shows the Loan Flow process modeled in Oracle BPEL Designer.

Once the process is deployed, the BPEL engine can provide metrics on the performance of each instance of a process. For example, Oracle BPEL Process Manager includes a BPEL Console that you can use for an aggregate view, or drill down into a particular instance of a business process's execution. However, you can also roll process metrics into a BAM solution. The BAM dashboard can be used to provide critical business data such as the number of loan requests approved during a certain timeframe, loan offers by provider, or the average processing time per loan. For example, Oracle BAM lets you instrument BPEL processes with sensors to monitor critical activities and variables. The BAM dashboard can also provide alerts – such as the approval of multiple customers with low credit scores or a loan rejection rate greater than 10% – that may indicate potential problems, and that let you take proactive steps. Figure 6 shows the BAM dashboard with key metrics for the loan flow process.

The optimization phase for the loan flow would involve using actual processing times based on weekly or monthly reports and then running the simulation tool again to generate possible bottlenecks and predict SLAs. This may result in a redesign of the process, such as adding new loan services to meet response-time requirements. Some of these metrics can also be used in real-time. For instance, if one loan agency takes too long to respond, the BPEL process flow can time-out and continue processing with only one loan offer to meet SLA requirements.



AppServer Type: JAVA J2EE	JAVA J2EE
Present AppServer	Tomcat 5.5.4
AppServer Virtual Path(s)/Extension(s)	/*.jsp /servlet/ /manager/ /*.do
:: Select Application Server ::	Tomcat 5.5.4

Localhost Bind Port Range	4671-4679
AJP/HTTP Bind Address	localhost:4671
Dedicated IP/Host Address	myaccount.webappcabaret.net
App Server Home Directory	/usr/ngasi/contexts/myaccount/e
Java Virtual Machine (JVM)	JDK1.5.0
Connector	AJP

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

Prices,plans, and terms subject to change without notice. Please log on to our website for the latest price and terms. Copyright © 1999-2005 WebAppShowcase • All rights reserved • Various trademarks held by their respective owners.

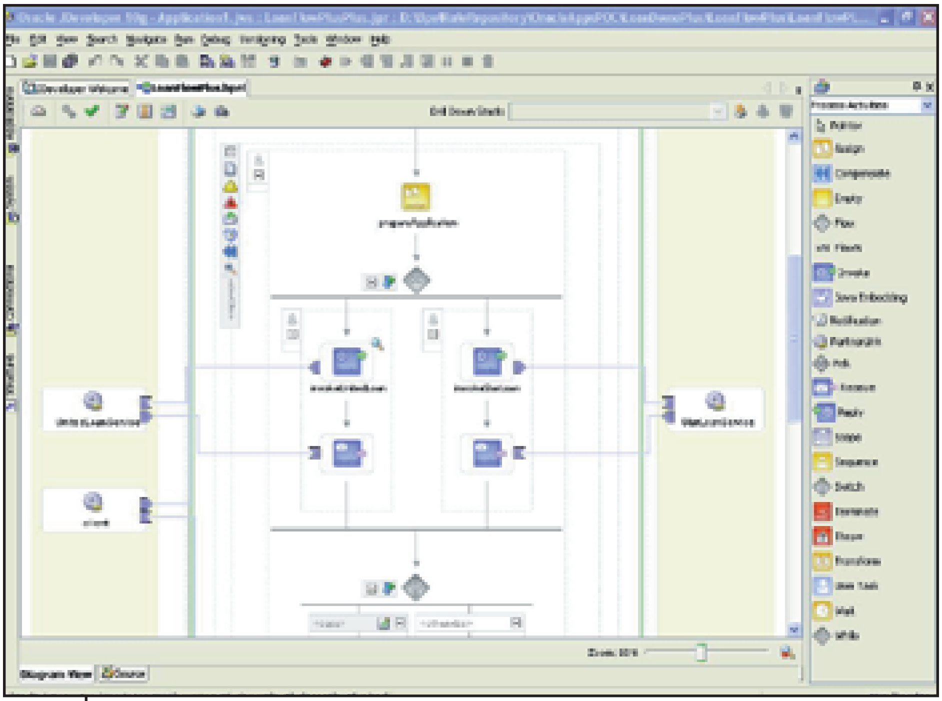


FIGURE 5 **Loan flow process design in Oracle BPEL Designer**

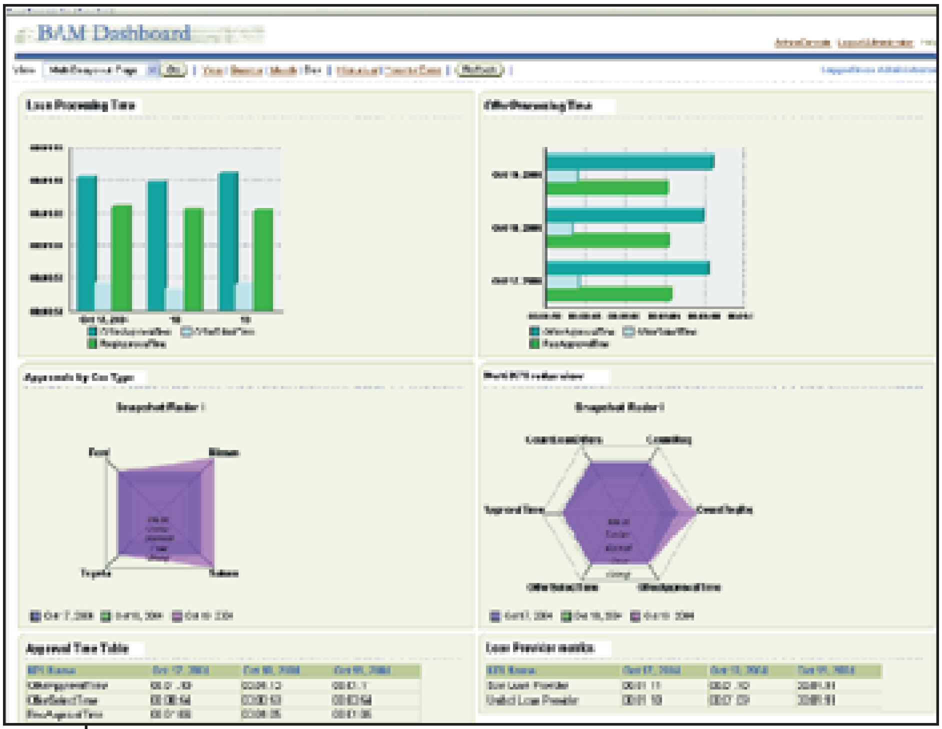


FIGURE 6 **BAM dashboard**

Any optimizations to the business process can be fed back to into the modeling tool, where changes can be made to the process using BPMN. Then the new process is exported to BPEL, which is picked up by the developer, who changes the implementation with BPEL and deploys it. Note that many process orchestration engines can run multiple definitions of the same business process concurrently. This is useful because when you change a business process, you may want existing instances to run through to completion on the old definition.

Conclusion

Closed-loop BPM systems are critical for developing efficient and effective business processes that can quickly respond to changing business conditions. A closed-loop BPM solution based on open standards lets organizations rapidly iterate over the complete process lifecycle using standards-based tools. One way to do this today is to use Popkin's tools to model and simulate processes with BPMN, export the definition to BPEL, implement the processes in Oracle BPEL Designer, and deploy them to Oracle BPEL Process Manager.

References

- *For more information on BPEL and closed-loop BPM:* http://otn.oracle.com/bpel.
- *For more information on Popkin and process modeling:* www.popkin.com.
- *Business Process Management: The Third Wave by Howard Smith and Peter Fingar (Meghan-Kiffer Press, 2003)*

Acknowledgements

The authors would like to thank Jon Dart, Frank Knifsend, and David Shaffer for reviewing this paper.

About the Authors

Bhagat Nainani is a product development manager in the Oracle Application Server Division. He currently leads the development of BPM services for the Oracle BPEL Process Manager. He has more than 10 years of experience with distributed systems, enterprise software, and integration technologies.

■ ■ ■ bhagat.nainani@oracle.com

Mohamad Afshar, PhD, is director of product management for Oracle Application Server 10g. He has core product management responsibilities for the Application Server. His main focus includes middleware vision, strategy, and architecture, with an emphasis on Web Services, SOA, and EDA. Mohamad is a frequent speaker at industry events and a contributing author to business-oriented, technical, and academic journals. He has a doctorate in parallel database systems from Cambridge University.

■ ■ ■ mohamad.afshar@oracle.com

Jog Raj is senior BPM consultant at Popkin Software. Jog was involved in the early development of the BPMN (Business Process Modeling Notation) at www.bpmi.org. He was also responsible for developing the mapping of BPMN-to-BPEL in System Architect. Jog has been involved in modeling business processes for over five years. He has been instrumental in the success of many blue chip client projects in both the U.S. and Europe.

■ ■ ■ jog.raj@popkin.co.uk

XSLmaker

XSL XML CSS Xpath

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

Skyway Builder from Skyway Software

Reaching for the sky

Services Oriented Architectures (SOA) rarely start from scratch. In most enterprises, they are built in gradual steps as part of an overall migration and architectural strategy. Along the way, existing legacy systems must be enhanced to support the required interface technologies, service gaps have to be filled by building new systems or acquiring and leveraging external services through third-party providers, and all elements in the architecture must be orchestrated and combined to form the final catalog of services. One solution that helps address these challenges is Skyway Builder from Skyway Software.

WRITTEN BY
BRIAN BARBASH

Skyway Software takes a three-layer approach to SOAs: Produce, Manage, Extend. The Builder tool is the Produce Layer and provides the capabilities required to build, integrate, orchestrate, and deploy composite applications in an enterprise.

Working with Builder

The first thing developers will notice about Skyway Builder is the user interface. Traditionally, an IDE uses some form of file manager or explorer type of layout. Typically, a hierarchy of projects and objects is located in a tree-view from which components on which to work can be selected and opened. Skyway Builder uses a desktop paradigm (Figure 1). In this virtual desktop, there are two system objects, Control Panel and System, while each individual project is represented as a folder. The Control Panel offers access to basic system operations including Debug, Deploy, Import, and Export. The System folder offers access to Skyway Builder's configuration and administration components. While the paradigm takes a while to get used to, it's effective in developing the specific kinds of applications Skyway Builder creates.

Applications created with Skyway Builder consist of the following objects:

- Structures and Collections – These objects make up an application's object model
- Web Processes – Objects that can be used to build a Web-based user interface
- Processes – Defines services that manage control flow and business logic
- Web Services and Java Services – Touch points to external systems using Java and Web Services technologies

For purposes of this review, I will use these components to create a simple stock portfolio manager. To simulate a service-oriented architecture, the portfolio manager will serve mainly as a data store and rely on a set of external Web Services for data validation, pricing, and other information.

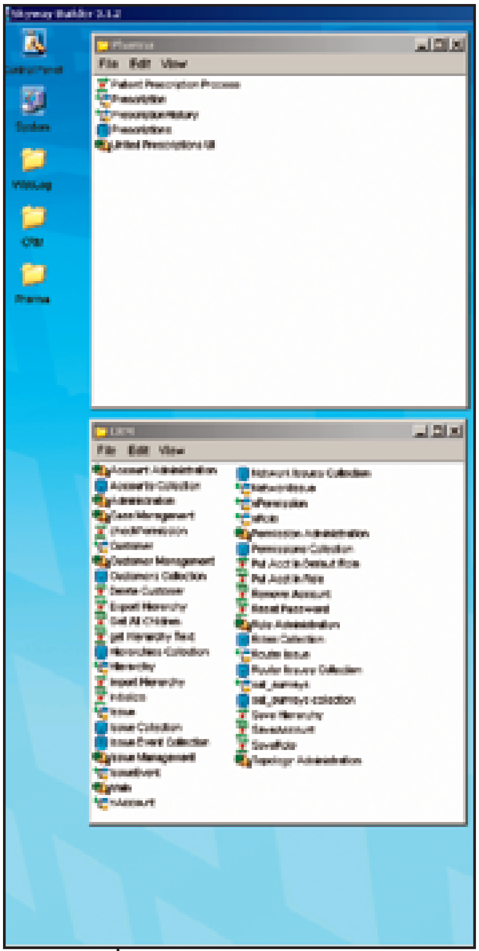


FIGURE 1 Skyway Builder IDE

Company Info
Skyway Software
208 S. Hoover Blvd
Suite 100
Tampa, FL 33609
Phone: 813-288-9355
Fax: 813-288-8266
www.skywaysoftware.com

Modeling Data

Central to any system or application is the data model. Skyway's Structures and Collections components build an object model. Structures are analogous to a class or JavaBean, while Collections are the containers in which multiple instances of a Structure reside at runtime. Skyway Builder allows structures to be created using a wizard, or to be imported from an external Web Service, Java Service, XML Schema, or data source.

In this example, data objects are required for all Web Services to execute. Skyway Builder can import a WSDL file from a remote URL or local filesystem. Three services are used in the portfolio manager: a locally hosted service for storing stock and transactional data, and two external services for getting the latest price and market data.

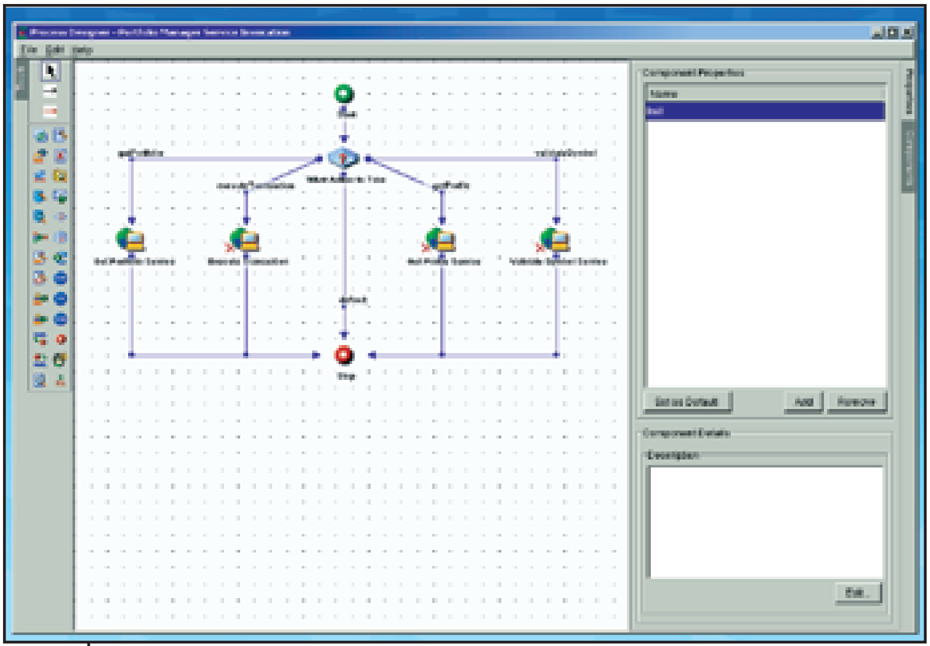


FIGURE 2 **Process Designer**

Integrating with Services

As mentioned above, Skyway Builder provides a way to import service definitions from WSDL files and Java classes. To execute methods from one or more Java classes in a JAR file, all dependent JARs must be in both the Skyway Builder client classpath and the Skyway Server runtime classpath. For our purposes, only Web Services will be executed.

The standard Skyway Builder Process object provides the entry point to all Web Service invocations and the mechanism for controlling application flow. Figure 2 shows the Process object designed to handle calls to local Web Services from within the Portfolio Manager application.

A decision tree exists at the entry point of the process that determines which Web Service/Operation combination to execute based on an action variable. Since the Web Services executed in this process are document-style interfaces from the same provider, the serviceInput variable is passed to each and the results collected in the serviceOutput variable.

Skyway Builder's Process objects contain a wide array of additional functionality to manage control flow, data manipulation, integration, and error handling so developers can create very complex logic flows. And any Process in Skyway Builder can easily be exposed as a Web Service simply by setting a flag in the Process settings. This flexibility lets Processes be shared across an organization.

User Interfaces

The user experience of composite applications in an SOA should be no different than their "stove-piped" counterparts. However, the back-end design and architecture of these interfaces can be. Skyway Builder brings to the UI a componentized approach well-suited to SOAs called Web Processes. A Web Process consists of one or more Web Components and possibly a collection of Events. Individual Web Components define a series of fine-grained steps. Examples include collecting and laying out HTML elements, modifying session and local variables, and executing and gathering data from other processes.

Web Processes, like their Process counterparts, are based on a flow and created in a similar designer. In the portfolio manager, an example Web Process for entering a transaction executes the following steps:

1. Accept input parameters
2. Validate stock symbol externally
3. Check latest price and calculate total cost
4. Add the transaction to the Portfolio Manager
5. Return confirmation

With the steps defined, the developer can focus on creating the HTML to display the UI itself. Skyway Builder provides an internal HTML authoring tool that developers use to access the structures, collections, and variables, as well as invoke the actions defined in a Web Process.

Summary

Skyway Builder from Skyway Software provides a solid collection of tools for working in a Services Oriented Architecture. Its focus is on the composition of higher-level services from existing fine-grained services. While the GUI paradigm isn't traditional, the transition to the virtual desktop concept is smooth and it proves to be effective in this context. Overall, Skyway Builder is a solid foundation for building services-oriented architectures.

About the Author

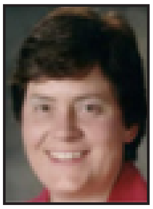
Brian R. Barbash is the product review editor for *Web Services Journal.* He is a senior consultant and technical architect for the Envision Consulting Group, a management consulting company focusing on contracting, pricing, and account management in the pharmaceutical industry.

■ ■ ■ bbarbash@sys-con.com

The State of Standards

Do they help us do our job, or do they just get in the way?

There is an old saying among standards wonks: "The most wonderful thing about standards is that there are so many of them." And this truism is more applicable today than ever before. There are so many WS-* specifications, I've started referring to them as WS-Vertigo.



WRITTEN BY
ANNE THOMAS MANES

But there is a reason that there are so many of them. The Web Services Framework (WSF) relies on a composable architecture. One of the primary tenets of the WSF is to keep things as simple as possible. Therefore, if an application doesn't require security, reliability, or transactions, you shouldn't clutter up the infrastructure with these capabilities. But at the same time, if the application does require security, reliability, and/or transactions, then you should be able to add support for the specific set of capabilities you need.

Refactoring Specifications

In order to enable a pick-and-choose, plug-and-play infrastructure environment, the WSF architects approached the problem using refactoring techniques. Refactoring is a way of restructuring a software system to make it clearer, cleaner, simpler, and more elegant. The basic idea is that if you have two components within a system that do the same thing, you want to refactor the duplicate functionality into a separate component. (See Refactoring: Improving the Design of Existing Code by Martin Fowler, ISBN 0201485672, for more information on refactoring.)

The WSF architects have refactored infrastructure functionality into its most basic constituent parts, and as a result, there is a different specification for each bit of functionality. Figure 1 shows an overview of the WSF architecture, depicting a set of broad functional components. There are three categories of functionality: the core, the extensions, and management, represented by the colors blue, pink, and green, respectively. The core provides the foundation for the framework, the extensions provide pluggable infrastructure functionality, and the management components enable control of the environment. Figure 2 maps a number of WS-Vertigo specifications to each functional component.

Notice that many of the functional components support a variety of capabilities, and each capability requires a separate specification. For example, in the security component, there is a basic security framework (WS-Security), bindings for various types of security tokens (Username, X.509, SAML, REL, Kerberos, etc.), trust services for obtaining and exchanging tokens (WS-Trust), security session management (WS-SecureConversation), and security federation (WS-Federation). There are a bunch of other specifications related to security not listed in Figure 2 (such as SSL/TLS, XML Encryption, XML Signature, XKMS, XACML, etc.), but these other specification are general purpose specifications that apply to many different domains, not just to the WSF; so they aren't included in the WSF architecture. Nonetheless, the WSF security specifications often interact with or depend on these general-purpose security specifications, and unfortunately, at this stage in the game, a developer needs to be cognizant of most of them.

Also notice that in some cases, more than one specification addresses a particular capability. For example, there are three competing specifications for attachments:

- SOAP with Attachments (over MIME)
- WS-Attachments (over DIME)
- MTOM (over MIME)

The WS-I Basic Profile (BP) recommends using SOAP with Attachments (SwA), but unfortunately, .NET doesn't support SwA. Today, if you want to build interoperable Web services, you should use WS-Attachments. In the long run, the industry will switch over to MTOM, which is the attachment mechanism defined to work with SOAP 1.2, but currently very few products support either SOAP 1.2 or MTOM.

Other areas of contention include event notification (WS-Eventing versus WS-BaseNotification), reliable message delivery (WS-ReliableMessaging versus WS-Reliability), transaction management (WS-Transactions versus WS-Composite Application Framework), and service management (WS-Management versus WSDM).

To Be or Not to Be a Standard

Among all of the specifications that make up the WSF, only a handful of them are formal standards. In fact, the two most basic specifications in the WSF (SOAP 1.1 and WSDL 1.1) are not formal standards. Standardization is an important process because it involves analyzing, vetting, and testing the specification to ensure that it is consistent and that it can actually be implemented. The SOAP 1.1 and WSDL 1.1 specifications have not been through this vetting process, and it shows. These two specifications contain a number of

OSBC | OPEN SOURCE BUSINESS CONFERENCE

APRIL 5-6, 2005

THE WESTIN ST. FRANCIS, SAN FRANCISCO
osbc.com | THE BUSINESS OF OPEN SOURCE

osbc.com
THE BUSINESS OF OPEN SOURCE

OSBC is the only conference that focuses on the business of open source strategies and solutions and what they mean to your company. This April, join top level executives and industry luminaries who are paving the way in the Open Source arena.

LEARN to build sustainable, higher-margin Open Source businesses.

HEAR directly from business leaders about how to start and sustain a business on Open Source technology.

NETWORK with your peers and the be first to meet the hottest up and coming companies in the Open Source industry.

CORNERSTONE SPONSOR

Sun microsystems

CORPORATE SPONSORS

intel. Microsoft Novell. ca Computer Associates OSDL REALM optaros inconsistencies, ambiguities, and errors, which have beenthe source of many interoperability issues.

The WSF specifications that have been standardized include:

- W3C SOAP v1.2
- W3C Message Transmission Optimization Mechanism (MTOM) 2005
- W3C XML Schema v1
- OASIS Universal Description, Discovery and Integration (UDDI) v2 and v3
- W3C XPath v1 and XSL Transformation (XSLT) v1
- OASIS Web Services Security (WS-Security) 2004
- OASIS Web Services Reliability (WS-Reliability) v1.1
- OASIS Web Services for Remote Portlets (WSRP) v1.0
- OASIS Web Services Distributed Management (WSDM) v1.0

Standardization doesn't necessarily imply industry acceptance and adoption, though. Support for some of these specifications is pervasive, including XML Schema, UDDI v2, XPath, and XSLT. Some of the newer specifications aren't yet pervasive, but they have strong industry support, and we should see wide deployments within the next year. These specifications include UDDI v3, WS-Security, WSRP, and WSDM. SOAP v1.2 and MTOM also have strong industry support, but it will probably take more than a year before we see them supplanting SOAP v1.1 and SwA or DIME. On the other hand, WS-Reliability has almost no industry support. Most vendors are implementing support for WS-ReliableMessaging (a non-standard alternative from IBM and Microsoft) instead.

So What's a Developer to Do?

The plethora of WSF specifications is enough to make a developer's head spin. But the average developer doesn't really need to concern herself with each and every specification. Instead, I recommend that developers stick with the basics and follow the guidelines defined in the WS-I profiles.

A WS-I profile is not yet another specification. A WS-I profile takes an existing specification and goes through it with a fine-toothed comb and finds any possible ambiguity – any place where the specification says MAY or SHOULD rather than MUST – or any place where some important behavior isn't completely specified, and it then constrains the specification and says, "Do it this way."

This type of profiling is particularly important when dealing with non-standardized specifications, such as SOAP 1.1 and WSDL 1.1, because these specifications have a lot of ambiguities. However it's also necessary with more formal standards, such as WS-Security, which offer a broad spectrum of options and choices, or with UDDI, which provides a basic registry, but doesn't define a standard model for registering services. The WS-I Basic Profile (BP) says register services this way. The WS-I Basic Security Profile (BSP) says secure your services using this constrained set of options.

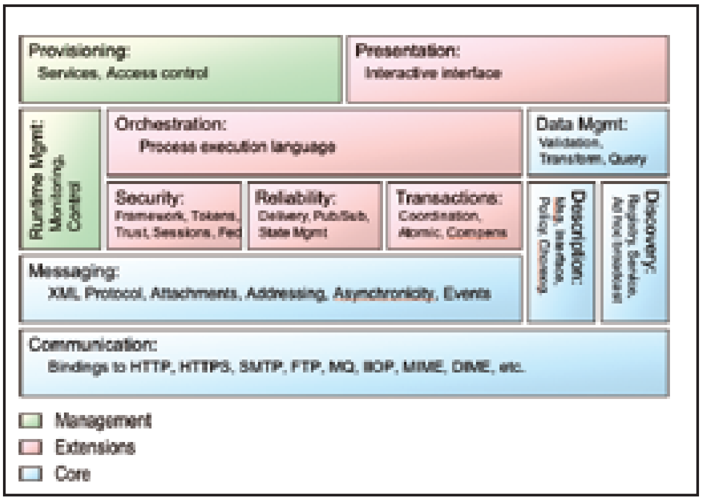

Provisioning:
Services, Access control
Presentation:
Interactive interface
Runtime Mgmt:
Monitoring, Control
Orchestration:
Process execution language
Data Mgmt:
Validation, Transform, Query
Security:
Framework, Tokens, Trust, Sessions, Fed
Reliability:
Delivery, Pub/Sub, State Mgmt
Transactions:
Coordination, Atomic, Compens
Description:
Msg, Interface, Policy Choreog
Discovery:
Registry, Service, Ad hoc broadcast
Messaging:
XML Protocol, Attachments, Addressing, Asynchronicity, Events
Communication:
Bindings to HTTP, HTTPS, SMTP, FTP, MQ, IIOP, MIME, DIME, etc.
Management
Extensions
Core

FIGURE 1 An overview of the WSF architecture

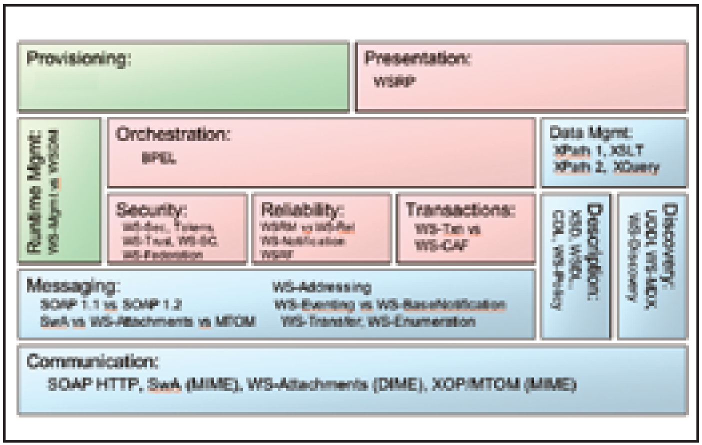


FIGURE 2 WS-Vertigo specifications and their functional components

Best Practices

Developers should also follow some basic best practices when building Web services. One of the most important concepts to understand about the WSF is that it is not a distributed object system. Web services communicate by exchanging messages – it's more like JMS than RMI. The WSF doesn't support remote references, remote object garbage collection, or any of the other distributed object features developers have come to rely upon in RMI, CORBA, or DCOM. The fundamental purpose of the WSF is to enable interoperability across dissimilar systems that don't necessarily understand concepts such as method overloading, object inheritance, and polymorphism. Hence Web service interfaces should not expose these OO concepts.

A service should expose a document-oriented interface rather

than an object-oriented interface. The basic best practices are:

- Flatten object graphs. Don't expose language-specific object collections, such as maps, lists, or datasets. Instead convert all collections into arrays.
- Don't use overloaded methods. Each method should have a different operation name.
- Expose a "chunky" interface rather than a "chatty" interface. In other words, don't expose getter and setter operations for every member in the object class.

These best practices relate back to the differentiation between message exchange versus distributed object systems. When using distributed objects, the object resides on the server side, and the client invokes operations on the object using a proxy. The client does not have its own copy of the object. When using a message exchange system, the client side application should have its own object – not just a proxy. (And – by the way – the client's object may be different from the server's object.) When the client communicates with the server, it simply passes data, not behavior. It's much more loosely coupled.

It may be necessary to build an abstraction layer between the WSDL interface and the application that implements the service. This abstraction layer performs the necessary mapping between the document-oriented WSDL interface and the application's object model. It also provides much better insulation for flexibility and change.

For best interoperability, developers should use document/literal with the "wrapped" programming convention. .NET uses the "wrapped" style by default. The JAX-RPC specification also requires support for the "wrapped" style. The "wrapped" style supports a programming model that makes document/literal feel like RPC style. "Wrapped" style is very similar to RPC/literal, except for two important distinctions:

1. .NET supports "wrapped" style, but it doesn't support RPC/literal
2. "Wrapped" style defines a schema of the full soap body (which makes it very easy to validate), while RPC/Literal only defines type information rather that full element schemas (which makes validation slightly more complicated).

Please see my blog entry for a definition of the "wrapped" style: http://atmanes.blogspot.com/2005/03/wrapped-documentliteral-convention.html.

What About Advanced Features?

For the moment, interoperability is still very challenging when using advanced features. Basic security is progressing well, but standards for reliable messaging and transactions are still in flux.

Today, most developers rely on transport-level services to support advanced features. For example, a developer can use HTTP Basic Authentication and SSL to implement Web services security. These transport-level security capabilities are typically sufficient for point-to-point communications. Transport-level security doesn't provide sufficient protection, though, when messages need to be routed to multiple services or through one or more intermediaries. In these situations, developers should use message-level security based on WS-Security. Most vendors have added support for WS-Security to the latest versions of their products. Developers can also use a software-based Web services management (WSM) product or a hardware-based XML gateway product to automate the use of WS-Security. Developers should follow the guidelines defined in the WS-I Basic Security Profile (BSP) for using both transport-level and message-level security mechanisms.

For reliable messaging, most developers currently use a message queuing transport, such as IBM WebSphereMQ or SonicMQ. These systems support guaranteed message delivery, but they sacrifice pervasive communications. The message queuing system must be deployed on both sending and receiving nodes. Eventually, the WSF will support reliable messaging over HTTP based on WS-ReliableMessaging (WSRM). A handful of vendors (Blue Titan, Cape Clear, and Systinet) currently support WSRM, and many other vendors, including BEA, IBM, Microsoft, Sonic Software, and TIBCO, have implementations of WSRM in the lab. These vendors should release WSRM products before the end of 2005, and reliability should become a pervasive capability in the WSF within about eighteen months.

Transactions are a bit stickier. My general recommendation is not to use transactions with loosely coupled systems. Instead design the application to use reliable messaging to coordinate interdependent services. Nonetheless, some applications may require two phase commit (2PC) transaction integrity. IBM and IONA currently support WS-Transaction within their Web services platforms, and Arjuna and Choreology provide third party transaction coordinators that support WS-Transaction. A handful of other vendors have implementations of WS-Transaction in the lab, and we should see more product releases in 2006.

Transparent Infrastructure Nirvana

Today, if you want to use advanced features, you need to understand all of the associated WS-Vertigo specifications, but in the long run, this won't be the case. Once the WSF matures and these advanced capabilities become pervasive, the tooling will manage the infrastructure functionality automatically. Play with Microsoft's "Indigo" to get a sense of what it will be like. With "Indigo" a developer simply annotates her code to indicate that she wants to use security or reliable messaging. The "Indigo" compiler then automagically generates the appropriate run-time code and policies that ensure that the Web service uses WS-Security and WSRM. The Java community also plans to use code annotations in a similar way in the next release of JAX-RPC.

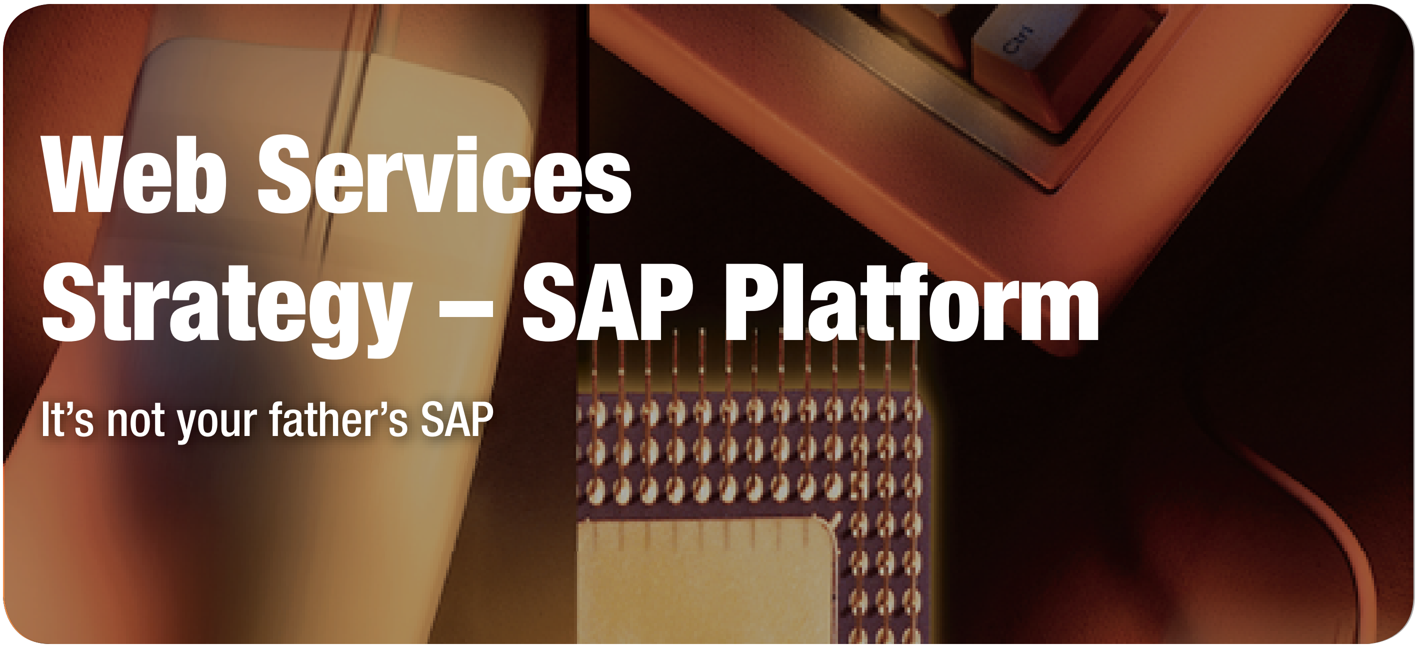
If you're a bleeding edge kind of developer, then have at it. Study the WS-Vertigo specifications to your heart's content. If you want to get your hands dirty, I suggest you play with the ws-fx projects being developed at Apache. But if you're just trying to get things done, then stick with the basics for the moment, and just don't worry about the WS-Vertigo specifications. The only specifications that you need to understand now are SOAP, WSDL, XML Schema, UDDI, and maybe WS-Security. And by all means follow the guidelines defined in the WS-I BP and BSP and stick with the best practices outlined above.

About the Author

Anne Thomas Manes is a Research Director at Burton Group, a research, consulting, and advisory firm. Anne leads research for the Application Platform Strategies service. Named one of NetworkWorld's "50 Most Powerful People in Networking," in 2002 and one of Enterprise Systems Journal's "Power 100 IT Leaders," in 2001, Anne is a renowned technologist in the Web services space. Anne participates in standards development at W3C and OASIS. She is a member of the editorial board of Web Services Journal. She is a frequent speaker at trade shows and author of numerous articles and the book, Web Services: A Manager's Guide, published by Addison Wesley.

Prior to joining Burton Group, Anne was chief technology officer at Systinet, a Web services infrastructure company, and before that she pioneered Sun's Web services strategy. A 24-year industry veteran, Anne developed her expertise working at a number of the world's leading hardware and software companies. You can reach Anne via e-mail at anne@manes.net or through her Web site at www.bowlight.net.

■ ■ ■ amanes@burtongroup.com



Web Services Strategy – SAP Platform

It's not your father's SAP

Packaged business applications have dominated enterprise IT landscapes for over a decade. Now these products are undergoing major changes to segway into the world of Web Services. SAP has been one of the most aggressive companies in embracing this revolution. Its NetWeaver platform is an ambitious suite of integration technologies designed to morph SAP into enterprise SOAs. SAP unveiled a four-year roadmap for NetWeaver in 2003. This article looks at the platform at its halfway point as well as SAP's vision for creating a Web Services-based agile business process infrastructure.

WRITTEN BY
SCOTT CAMPBELL

What Is NetWeaver?

NetWeaver is really a brand that encompasses a suite of tools just like WebSphere, WebLogic, and .NET are umbrellas for a family of products within a platform. Some of these products are relatively new, while others are proven solutions that have been widely deployed. What is significant is the way SAP is bringing these products together in support of a common vision for service-oriented computing. This has led to a unified roadmap, better compatibility, and shared release cycles for the entire suite.

All of the usual suspects can be found inside of NetWeaver. First is an application server that SAP calls the Web Application Server or WAS. What's unique about WAS is that it contains both a fully compliant J2EE engine and SAP's ABAP engine, which serves as the technical foundation for most SAP products. In fact one might argue that SAP figured out the application server concept long before Java. And much of the proven high-availability, security, and fault-tolerant technology built for the ABAP environment has been extended to the Java engine. WAS also contains rich support for exposing both Java and ABAP-based applications as Web Services as well as consuming external Web Services.

The next component is an integration broker and business process management (BPM) tool called Exchange Infrastructure or XI for short. XI offers the same modeling, design repository, and execution environments you'd expect from any EAI tool. Besides traditional adapter models it was designed with XML-, SOAP-, and Web Service-based integrations in mind. The BPM part of XI offers a fully complaint BPEL orchestration modeler and execution engine. Table 1 provides a more complete view of how XI supports Web Services standards.

SAP will soon begin to ship significant amounts of pre-built integration content modeled in XI along with its business applications. What's unique about XI is the way SAP is trying to bridge the gap between coarse-grained business process definitions that users understand and the much finer-grained orchestration execution models that exist today in BPEL semantics. SAP is partnering closely with IDS Scheer, a leading business process automation firm, in this endeavor. Closing that gap is a prerequisite to realizing the promise of agile business-driven computing.

The third major component of NetWeaver is a portal server that SAP calls Enterprise Portal. As with any portal server Enterprise

Portal acts as an integrated user interface across many applications. It offers role- and user-based personalization, single sign-on APIs, a portlet model called iViews, XML/XSLT data transformation support, and so on. Enterprise Portal also includes tools for easily exposing portal services as Web Services.

NetWeaver contains many other products including workflow and collaboration tools, a private UDDI registry, a business intelligence suite called Business Warehouse, a Mobile Infrastructure Engine for accessing SAP from portable devices, and Master Data Management for synchronizing and harmonizing data. Accompanying these products is a set of modeling, development, deployment, and monitoring tools, some of which will be discussed in more detail below.

In short NetWeaver is a technology platform with many of the features you would expect from any application development and integration middleware stack. In addition, Web Services are becoming a pervasive part of this platform. NetWeaver will be part of every SAP system landscape in the future because it's part of the upgrade path. Enterprise Portal will become the ubiquitous user interface across the company's business suite. XI will be the out-of-the-box way to integrate SAP with other applications. Most importantly, SAP customers will soon see a flood of new Web Services options unleashed on their organizations.

How XI supports Web Services standards

Application & Industry-Specific Services	(UBL, xCBL, HRXML, GCI, RosettaNet, ...)
Service Publication & Discovery	(UDDI, ebXML Registry, ...)
Service Choreography	(WSCI, ebXML BPSS, ...)
Service Description	(WSDL, BPML, BPEL, ...)
Service Invocation	(SOAP, ebXML Messaging, RNIF, ...)

NetWeaver Is About Business Processes

At first glance it looks like SAP is going head-to-head with IBM or BEA in application middleware. While there's a lot of feature overlap between these platforms that's not the intent. SAP is first and foremost a business applications company. NetWeaver is designed to morph SAP's business applications into the SOA world. The reason is simple. It's inevitable that process logic will be liberated from individual applications.

Let's reinforce the point. What it means is that the promise of Web Services, loosely coupled architectures, and digitized business process management is about to enter a new age. Companies like SAP that have been built on monolithic products with proprietary interfaces and high switching costs have surrendered to this eventuality. By embracing the SOA paradigm companies like SAP will actually accelerate the adoption curve.

SAP's vision for service-ori-

IN THE NEXT ISSUE OF *WSJ...*

FOCUS: Financial Industry

Computational XSLT for Financial Statements

Schema languages like XBRL (Extensible Business Reporting Language) can define the structure of a financial statement, and the data itself can be saved as an XML instance of the schema. The data is often processed further using formulas; for example, to verify balances and derive data for financial analysis. This processing is traditionally done in a non-XML application, but staying in the world of XML, I ask: How well can XSLT (Extensible Stylesheet Transformation Language) do these calculations and what are the advantages?

BizTalk Server 2004: Too Hot to Handle?

Recent trends in IT such as the Service-oriented architecture and Web services, in conjunction with the still-increasing popularity of the .NET framework, put Microsoft's BizTalk Server in the center of attention for CIOs, CTOs, architects, and enterprise developers. It appears that everyone who does something with the Microsoft platform now wants to implement BizTalk.

Effective Business Process Management (BPM) Software

Conventional wisdom holds that you can't implement business processes involving human worker interaction and integrations with enterprise applications without a heavy reliance on IT resources. Throwing requirements "over the wall" to IT is where the breakdown occurs. How many CRM implementations have jury-rigged "workarounds" because necessary components proved to be "out of scope" – an issue too often discovered post roll out?

Coordination and Transactions Are Key to Building an Operational SOA

When building an SOA, most are leveraging loosely coupled type architecture. The benefits of building a loosely coupled SOA with many services are apparent, though the operational characteristics could be a bit of a nightmare. However, with a bit of planning, and the use of some standards, your SOA will be as reliable as it is functional.

WebServices JOURNAL .NET J2EE XML

ented, process-driven computing is called the Enterprise Services Architecture (ESA). It is key to understanding how all the NetWeaver pieces come together. SAP defines ESA as “Service Oriented Architecture principles and Web Services technology applied to the world of enterprise business applications.” It means that value comes from fusing business applications with an enabling technology platform. ESA enables:

- Role-based UIs built on Enterprise Services
- Process automation in and across organizational boundaries built on Enterprise Services
- Decoupling process and integration logic

Implied in these definitions is the notion of an Enterprise Service with three characteristics. First, it’s a business-level service, which means that it’s fairly coarse grained and delivers value at a functional level. Second, it’s a service so it can be discovered, described, and invoked like any other Web Service. Finally, it’s enterprise-class implying high levels of scalability, security, manageability, and other non-functional concerns.

As with any SOA model these Enterprise Services become the focal point for a layered architecture. They’re built on a foundation of application and technology services that act as building blocks. Enterprise Services can then be woven together through composition techniques into applications that support cross-functional business processes. SAP has developed a commercialization strategy for these composite applications under the name xApps. xApps are business solutions that cut across multiple SAP and non-SAP applications using SOA integration patterns. SAP and its certified partners offer these composite applications as products you can buy. Examples include everything from pricing analytics to streamlined international trade management to optimizing exploration and production for oil companies. What’s significant is they are concrete business-level solutions being built, sold, and deployed under a services composition model.

Making NetWeaver Work

So far we’ve described SAP’s vision for service-oriented, process-driven computing called the Enterprise Services Architecture. We’ve also shown that SAP has a technology platform and suite of tools that lets us build and deploy composite applications. The next question is what lets us bring all of this together to make things happen? The following outlines a very basic evolutionary approach for leveraging the power of NetWeaver to build service-oriented applications and automate business processes.

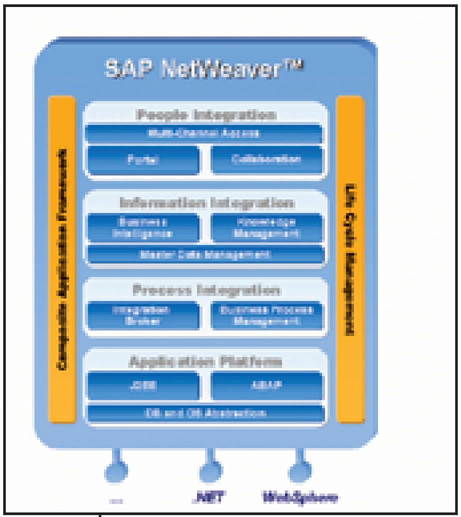


FIGURE 1 **Netweaver platform stack**

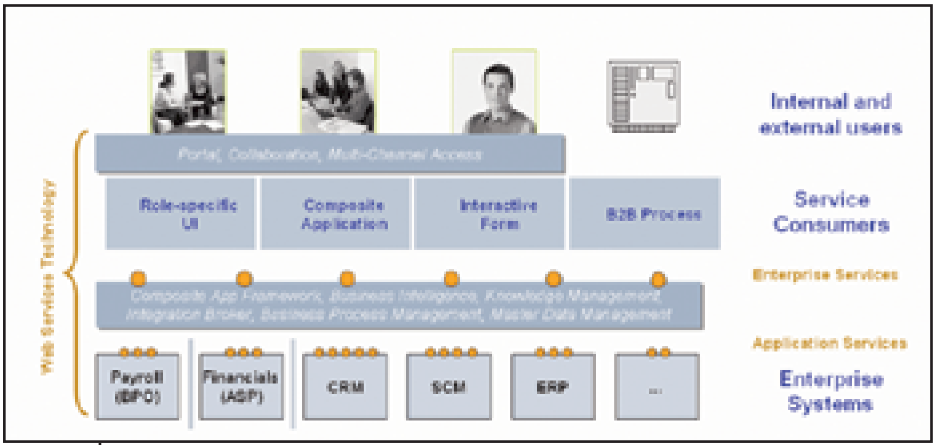

Internal and external users
Portal, Collaboration, Multi-Channel Access
Role-specific UI
Composite Application
Interactive Form
B2B Process
Service Consumers
Enterprise Services
Composite App Framework, Business Intelligence, Knowledge Management
Integration Broker, Business Process Management, Master Data Management
Application Services
Payroll (BPO)
Financials (ASP)
CRM
SCM
ERP
...
Enterprise Systems
Web Services Technology

FIGURE 2 **Enterprise services architecture**

Step 1: Start Using SAP’s Web Services

This is the obvious first step to realizing the ESA vision. In the layered architecture described above we begin with foundational technology and basic application interface-level services. These are our building blocks. Today all SAP applications can expose or consume Web Services through the Web Application Server or as endpoints modeled in Exchange Infrastructure. This means any .NET or Java developer can be an SAP developer from the standpoint of integrating with the SAP business applications. No longer do you have to know all the technical details around the ABAP, BAPI, iDoc, and RFC technologies that have made SAP applications so difficult to integrate in the past. Likewise, ABAP developers can now construct and expose Web Services using the Web Application Server. Essentially any out-of-the-box or custom SAP function module can be deployed as a Web Service.

As mentioned, these Web Services provide value at the basic application interface level, or as foundation technical-level services such as user management and security services. Examples might be “update invoice line item #2,” “convert dollars to euros,” or “determine if Jill can view the invoice.” NetWeaver also offers public and private UDDI servers for creating a directory of Web Service definitions. In addition there is logging and tracing support for Web Service invocations. Finally, the NetWeaver Developer Studio includes tools, wizards, and utilities to simplify the entire Web Services creation and deployment process.

Armed with this level of Web Services support developers can easily integrate SAP and non-SAP applications using their preferred development language.

Step 2: Leverage XI to Build Enhanced Services and Digital Process Execution Models

With a full complement of foundational Web Services in place, NetWeaver can be used

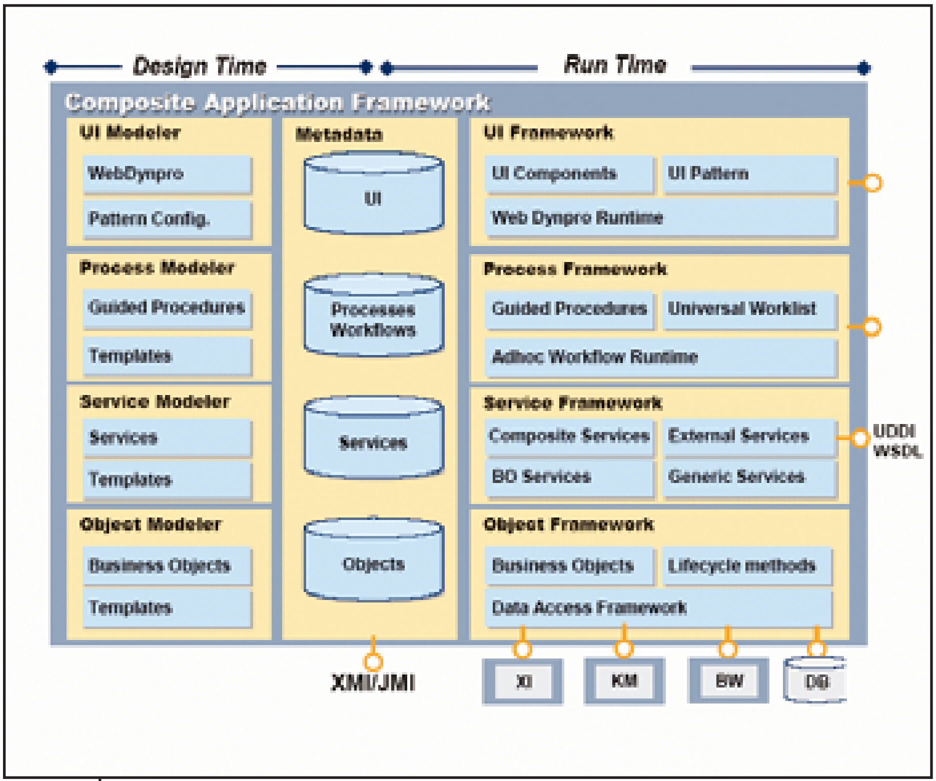


FIGURE 3 **Composite application framework**

to define more coarse-grained composite services. Exchange Infrastructure enables the kind of modeling and composition where services can be woven together to do larger units of work. More importantly, the compositions themselves can be exposed as Web Services for consumption by other applications. An example of services at this level could be "pay invoices" or "optimize purchase discounts."

The XI integration repository captures the design model. A complementary integration directory is used to capture the deployment model with specific information about system landscapes and endpoints. Finally, the integration engine controls the runtime behavior by executing service choreography in a stateful manner as well as providing real-time activity monitoring.

As mentioned above, SAP and IDS Scheer have partnered on tools to convert high-level process models developed by business analysts into digitized process execution models in XI. Expect this alignment to be significantly improved over time. The implication is that composite Web Services will more closely match real business process activities.

Step 3: Define Your Enterprise Services

In step 1 we identified our technology and application integration level building blocks and exposed them as Web Services. In step 2 we wove them together to create more coarse-grained Web Services that span functional and application boundaries to perform larger units of work. From a technical standpoint these are the steps needed to create the enterprise-grade services that are part of SAP's ESA vision.

However there are more than just technology considerations for having proven Enterprise Services. We've identified eight characteristics that make composite services valuable Enterprise Services. They are:

- Locatable – "I didn't know the service was available!"
- Reachable – "The service used the wrong protocols."
- Well Documented – "I couldn't figure out what the service did!"
- Usable – "The service was too narrowly defined for my business needs."
- Policy Based – "The service didn't support my non-functional policy requirements."
- Adaptable – "The service was too hard to change for my needs."
- Managed – "I didn't trust the service availability for my SLA."
- Encouraged – "I had no incentive to use the service."

SAP's NetWeaver roadmap is designed to address many of these needs. Later this year SAP will release its initial view of an Enterprise Services inventory based on its business applications. The focus will be on business process agility and this list can be used immediately for planning purposes. There will also be preview releases of an Enterprise Services Repository that will be another vital component of the NetWeaver platform. This repository will make Enterprise Services easier to locate, access, and understand.

SAP is also investing heavily in developer education, solution reference architectures, templates, and processes to teach developers how to be productive in an SOA environment. These assets will help developers leverage the Enterprise Services delivered with NetWeaver as well as plan and model new services based on their organization's unique business processes.

Step 4: Develop Composite Applications

By now you know that NetWeaver is a powerful application development platform. More importantly, its products and technologies create a Web Services-based enterprise SOA. But an SOA investment is only valuable when it's used to create cross-functional composite applications that solve real business problems in an agile way.

This new development approach creates challenges that aren't unique to SAP or NetWeaver. In a SOA world we define much of our business, application, and integration logic in metadata layers. This creates an interesting paradox. On the one hand, we are liberating integration and process logic from our business applications. This has been the promise of BPM hype for years. At the same time building applications and building integrations are becoming one and the same thing. Both involve leveraging loosely coupled Web Service invocations, just at different levels of granularity. That's what composition and SOA principals are all about. It breaks the traditional object-based design paradigms. We end up needing a whole new set of modeling artifacts and development tools to make best use of an SOA. This is where NetWeaver's Composite Application Framework (CAF) will come to the rescue.

The CAF enables the efficient, model-driven development of composite applications. CAF makes use of all of NetWeaver's components while hiding the complexity. Instead of having to move from tool to tool and build parts of your program in each product's environment, CAF provides a unified project. More importantly, it automatically generates significant amounts of code that saves developer time. The design time portion of the CAF involves modeling at four main layers. First are the entity services that define the business domain. Second are application services that capture business logic. Next is user-interface modeling that's role-based and can be composed from pre-existing UI patterns. Finally process and workflow logic are modeled.

WSJ Advertiser Index

Advertiser	URL	Phone	Page
EV1 Servers	www.ev1servers.net	800-504-SURF	6
Forum Systems	www.forumsys.com	801-313-4400	3
Global Knowledge	www.globalknowledge.com/lw	919-461-8600	33
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	Cover III
IBM	www.ibm.com/middleware/flexible		5
Illustro Systems International	www.illustro.com	866-4-illustro	Cover IV
ISSJ	www.ISSJournal.com	888-303-5282	36
IT Solutions Guide	www.sys-con.com	888-303-5252	45
Mindreef	www.mindreef.com/tryout	603-465-2204	9 & 11
NetWorld + Interop	www.interop.com	415-905-2300	35
OSBC	www.osbc.com		21
Parasoft	www.parasoft.com/products	888-305-0041	Cover II
Promind Systems	www.xslmaker.com	415-359 9117	17
Secrets of the XML Masters	www.sys-con.com/freecd	888-303-5282	57
SYS-CON Publications	www.sys-con.com/2001/sub.cfm	888-303-5282	46
Web Services Journal	www.wsj2.com	888-303-5252	51
WebAppCabaret	www.webappcabaret.com/jdj.jsp	866-256-7973	15
WebRenderer	www.webrenderer.com	61 3 6226 6274	39
XTech Conference	www.xtech.org		29

Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

These design time models are managed in a metadata repository that supplies execution code to the runtime engines of the various NetWeaver products. Version 1.0 of the CAF environment was released in 2004 and is limited. The next version will be more tightly coupled with SAP's WAS and provide improved modeling and support for all NetWeaver components.

A Dose of Reality

The NetWeaver platform has got a lot going for it, but it's not without its challenges. SAP is only halfway into the suite's planned development roadmap so right now it consists of equal parts "marketecture" and real product architecture. A lot of consolidation and integration needs to happen between the mature components and the newer products.

Second, SAP will always lag the market in adhering to the latest Web Services standards. A company whose products run on platforms ranging from the mainframe to Windows with an installed base of trillions of dollars of worth of transactions on its software has to be methodical in releasing highly reliable upgrades.

NetWeaver will also be hindered by the same technical problems that influence the rest of the Web Services, BPM, and SOA community. They include everything from the need for continued evolution of process modeling specifications to improved security, performance, and layering of technical policies. These are not SAP-specific challenges and NetWeaver's ESA strategy can only move as fast as the industry itself. Of course SAP along with IBM and Microsoft have had tremendous influence when it comes to driving these standards

Finally, NetWeaver will have to deal with user-adoption considerations. Relatively few IT professionals understand the SOA principals and value yet, which leads many SAP customers to perceive NetWeaver as merely as something they "have to deal with" as opposed to understanding the real opportunity for creating an agile IT landscape.

The best way to track the latest developments with the ESA vision and all of the NetWeaver products is on the SAP Developer Network Web site at http://sdn.sap.com.

Conclusion

NetWeaver is the technology platform on which all of SAP's business applications will be built. It's part of every SAP customer's upgrade path. At the halfway point in its implementation roadmap NetWeaver has become a great platform for exposing and consuming web services in SAP applications. It makes SAP products easier to integrate into non-SAP environments and includes significant support for Java and .NET development. NetWeaver currently provides the technology needed to model and orchestrate composite applications that execute cross-functional business processes.

The ultimate goal of NetWeaver is to provide the foundation for SAP's enterprise SOA strategy called the Enterprise Services Architecture. Because SAP is the largest business applications company in the world this has two implications. First, it validates that Web Service-oriented, process-driven models are here to stay because they help solve business problems through better agility and reuse. Second, SAP's product roadmap will accelerate the proliferation of Web Services computing and SOAs in many organizations that would otherwise not move to the new paradigm. These organizations will need education, training, and readiness planning for the Web Services world.

About the Author

Scott Campbell is a partner in Momentum SI, a systems integrator specializing in SOA-based technology strategy, integration, and development. He leads its NetWeaver integration practice, which helps organizations plan for SAP's ESA roadmap, leverage NetWeaver tools to build process driven composite applications, and mentors developers in working in Web Services- and SOA-based environments.

■ ■ ■ campbell@momentumsi.com

New for 2005! Two events are combining forces to bring attendees the most comprehensive group of web and business technology developers available.

XTECH CONFERENCE
XML, THE WEB AND BEYOND

THE GILBANE CONFERENCE ON CONTENT MANAGEMENT TECHNOLOGIES

24-27 May, 2005
Tutorials, 24 May • Exposition, 25-26 May
Amsterdam RAI Centre, Netherlands

Produced by:

IDEAlliance®

XTech 2005 is the premier European conference for developers and managers working with XML and web technologies.

Don't miss the expo featuring today's top web and business tech developers!

FOR COMPLETE DETAILS AND CONFERENCE PROGRAM SEE:

www.xtech.org

24-26 May, 2005
Tutorials, 24 May • Exposition, 25-26 May
Amsterdam RAI Centre, Netherlands

The Gilbane Conferences provide vendor neutral educational information for IT managers, system architects, and technically oriented business and project managers. This conference will cover:

Enterprise and web content management
•
Intranets, knowledge management and collaboration
•
Enterprise search, categorization and taxonomies
•
Document management and compliance
•
And many other technologies for creating, managing and delivering enterprise content

REGISTER TODAY! FOR MORE INFO SEE:

www.lighthouseseminars.com

Register for the Conference Plus Package and receive an **iPOD!**

Diamond Sponsor: JUSTSYSTEM

Gold Sponsors: Adobe · BLASTRADIUS XMetaL®

Media Sponsors: Business Wire *The Global Leader in News Distribution* · EUROPEAN TECH WIRE · THE GILBANE REPORT · mozilla FOUNDATION · O'REILLY® · WebServices JOURNAL

Enterprise Semantics: Aligning Service-Oriented Architecture with the Business

Bringing agility, transparency, and integration to your Web services infrastructure

Service-oriented architectures (SOAs) are a significant step forward in aligning information technology with business goals. But SOAs are insufficient when the Web services of which they are typically composed use inconsistent terminologies and present different understandings of the real-world environment that they are meant to support. Enterprises need transparency, a clear view of what is happening in the organization. They also need agility, which is the ability to respond quickly to changes in the internal and external environments. Finally, organizations require integration: the smooth interoperation of applications across organizational boundaries. Encoding business concepts in a formal semantic model helps to achieve these goals and also results in additional corollary benefits. This semantic model serves as a focal point and enables automated discovery and transformation services in an organization.

WRITTEN BY
JOSHUA FOX & JORAM BORENSTEIN

Technology and the Enterprise

Enterprise Information Technology (IT) units exist to support organizational business goals. They must always be aware of the need for alignment, the challenging task of ensuring that every IT decision, in every department, helps the business side of the organization to get its job done, and requires constant monitoring of IT assets to be sure that they correctly describe the organization as the organization understands itself. To this end, it is not enough to follow the common industry practice of creating new data sources or layering new applications on top of existing data sources. Information sources and the applications tying into these sources must function in a smooth and unified manner, thereby allowing units and individuals to work together. When aligned, all stakeholders can coordinate efforts both within an organization (internally) and across organizational boundaries (externally).

Services-oriented architectures are playing an essential role in the drive to integrate information and applications across the enterprise. With SOAs in place, technical teams respond quickly to provide a collection of applications that function as a clearly understandable whole. In this way, they provide a powerful tool to the rest of the organization; in fact, such an integrated process can create as much value as all the existing unintegrated processes.

However, alignment is impossible when

business departments misunderstand one another and when IT misunderstands the business. All the XML and HTTP in the world cannot integrate applications attempting to communicate in different terminologies. To ensure accurate communication, every application must be developed in strict accordance with the same standard (an impossibility in the real world), or else information managers must analyze the business concepts underlying the data and the applications, relate these concepts to existing industry and enterprise standards, and then encode that analysis in discovery and transformation logic, hence bridging the barriers. Today, this analysis must be repeated on every project. Even then, it remains painfully error-prone. With this in mind, Gartner recommends that enterprises "ensure that all developers use the same semantics and agree on the message contents at a business level." (See "SOA and Web Services Offer Little Vendor Independence," in Resources.)

Service-Oriented Architectures

"Service-oriented architecture" is among the latest three-letter acronyms to hit the technology world. But buzzword cynics should know: "SOA" does have a real meaning. The SOA approach is to expose enterprise data and business logic in loosely coupled, discoverable, structured, standards-based, coarse-grained, stateless units of functionality (see "W3C Web Services Architecture Working Group" in Resources). These are defined as follows:

- Loosely Coupled. Applications talk to one other and to other databases – without regard for implementation language.
- Discoverable. A Web service's network location is not predetermined and can be looked up at run-time.
- Structured. Messages have an interface (schema) that defines their structure and which can also be discovered. This allows applications to read messages from other applications and data sources without the service provider needing to define the exact message-deserialization code.
- Standards-Based. The message format and schemas are known to all parties.
- Coarse Grained. Each interaction should provide information or functionality achieving real business value; multiple exchanges of information should not be needed for each step in a business process.
- Stateless. Though not an absolute requirement, it is better that each message does not change the state of the application – though data can be stored. This allows lost messages to be resent without complicated transaction-processing logic.

In practice, SOAs are achieved with the Web services standards stack of XML, XSD, WSDL, SOAP, and UDDI; XML for the message format, XSD for the message structure, WSDL to tie together information on the service, and UDDI for service discovery. In addition, vertical-specific standards constrain the schemas of the messages sent.

The goal of smooth integration is not achieved by SOA alone. Web services must be described and understood in a semantically consistent way in order to resolve terminological ambiguities and misunderstandings, and to avoid the constant revision and redefinition of terms, concepts, and elements of the business. Such inconsistencies make for applications that cannot talk to each other, and subsequently result in slower response times when changes are needed. Business managers cannot get a clear view of their organization through these multiple unintegrated "languages."

To take a simple example, a manufacturing report service may indicate that one million units of a product were manufactured, while a shipping report service states that only 100,000 were shipped. Extensive analysis might be needed to determine that the shipping report is designed to report the number of packages shipped, each of which includes ten of the original manufactured units. If the former report is for a week measured Monday to Monday, and includes all shippable and defective products, while the latter is for Sunday to Sunday, and excludes defective items and returns, more confusion ensues.

> "Services-oriented architectures are playing an essential role in the drive to integrate information and applications across the enterprise"

Semantic Service-Oriented Architecture

In a semantically based SOA, an information model expresses a common linguistic understanding of the real-world business environment across individual organizations or even across entire industries. This allows executives, analysts, data specialists, and application developers to share a common understanding of concepts relevant to the business, a partnership, or an entire vertical industry. According to Gartner, "Data semantics becomes a critical barrier for SOA as more participants involve more applications." (See "SOA Will Demand Re-engineering of Business Applications" in Resources.)

In the shipping example above, the semantic model includes concepts such as "Manufactured Unit," "Package," "Date," Week," "Shippable/Defective," and the relationship between such concepts; it also relates to the manufacturing and shipping report services themselves. Business rules show the relationship between the different ways of expressing the same concepts. In this way, the meaning of the Web services is formally expressed through relation to an agreed upon set of business concepts.

In terms of metadata, SOA maintenance means that development teams require an inventory of services, back-end systems, inter-service interactions, and different service versions. From an architectural standpoint, the Web service is only as good as its interface as described in Web Service Description Language (WSDL). To create useful Web services, enterprises require a methodology for modeling their common business language and for inferring consistent WSDLs. To maintain accurate interrelationships, Web services must be mapped to the back-end applications they expose, and to each other as the business and its trading partners evolve incompatible WSDL schemas. In the semantic SOA, organizations manage their disparate WSDLs, which are front ends for different data assets, IT systems, business processes, networks, and applications, by reference to a common business architecture expressed and not in implementation-oriented terms, but rather according to the needs of the business.

Technical Background

From a technical perspective, the information model's ability to accurately express the common understanding of services is achieved with the following process:

1. The semantic information model is built, often on the basis of an industry-standard language, such as ACORD for insurance, HL7 for health care, or FIX for financial services.
2. The relevant services are identified.
3. Each service interface or schema is mapped to the relevant concepts in the semantic model. This mapping starts off coarsely grained, giving proper semantic meaning to the service message as a whole (or to major message components) and proceeds to a fine granularity, assigning meaning to every field in its proper context. The flexibility inherent in these mappings allows the model's stakeholders to agree on the precise meaning of the services involved, thus eliminating the confusion caused by heterogeneous interfaces.

This is an incremental process. Identifying services and mapping them to the semantic model generally facilitates the understanding of how the business works, thereby enhancing the semantic model.

Each individual Web service typically has its own language, expressed in a schema, for input and output. When these languages are integrated through a semantic model, users get a coherent understanding of the schemas, which provides maximal transparency, integration, and IT infrastructural agility. The semantic model allows flexibility in updates to a Web service interface flexibly – clients can "understand" the changes with ease, given that the differences between schemas have been resolved by reference to an information model. When a client requires a service, the service that satisfies the business need can be discovered by reference to the semantic information model, with no need to predetermine the exact service interface. When a client requires input that is incompatible with a given service's output, then translation scripts transforming between the two are an automatic product of the semantic model and mappings, having been inferred from the semantic relationship established previously through the semantic model.

Aligning IT and the Business

IT staff understand the technical value of greater transparency, agility, and integration, but often fail to connect these to the ultimate needs of the enterprise, namely

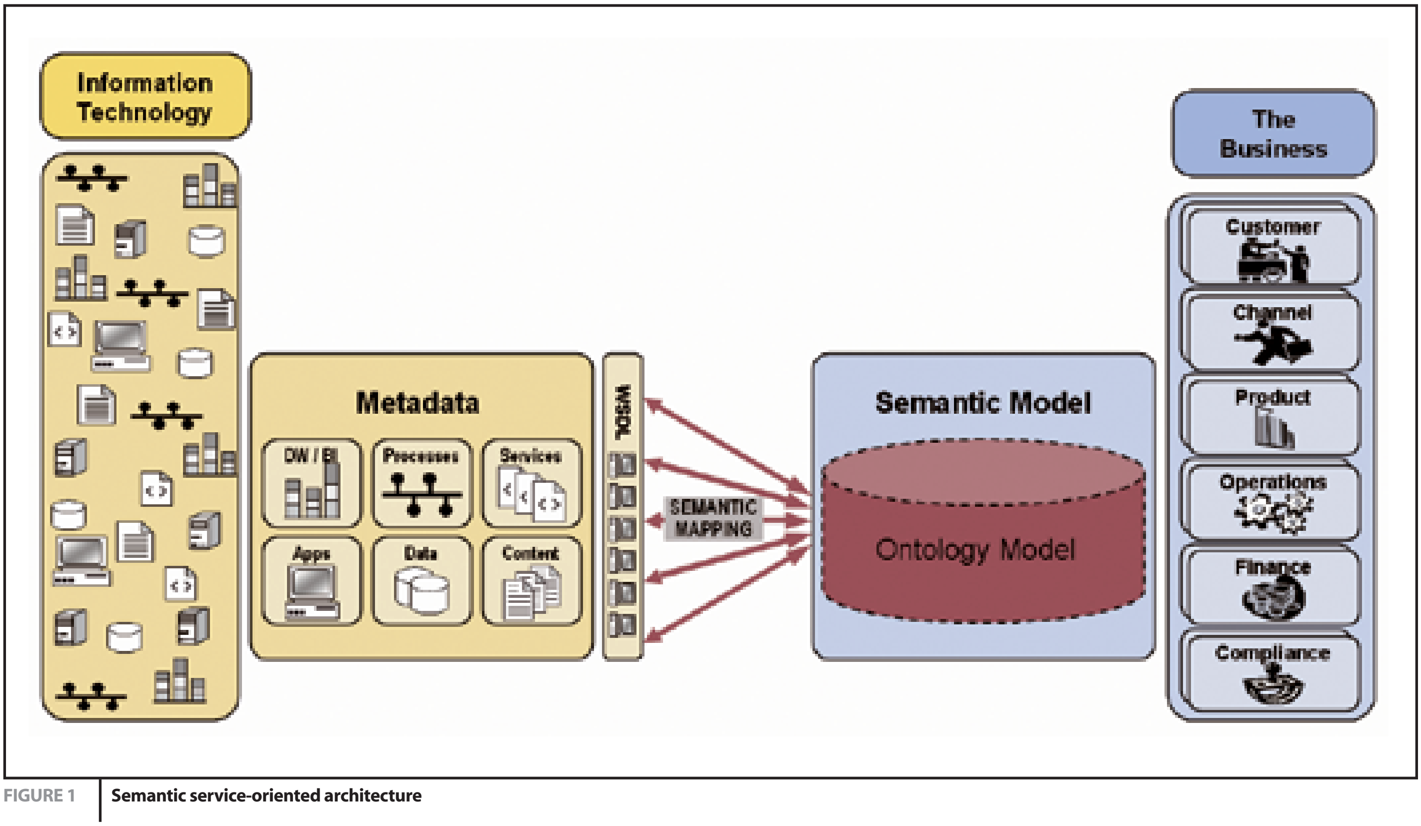


FIGURE 1 Semantic service-oriented architecture

Are you Red Hat® ready?

creating value according to the goals of the organization. Semantics as added to SOAs, however, tie the technical achievements tightly to the business context.

Transparency

Transparency is critical to modern organizations. Sarbanes-Oxley legislation, Basel II restrictions, and other accounting and privacy regulations in various countries make it essential to expose the true meaning of information storage and transfer to the watchful eyes of internal and external auditors.

Transparency in the enterprise is defined as the ability to trace and to understand the meaning behind enterprise data and enterprise assets, an ability increasingly important to business intelligence and executive reporting. Without a consensual understanding of a given business term, ambiguity and confusion reign. Semantics ensure that Web services and resulting user-facing reports use an unambiguous business language, and that all messages and reports can be traced back to their data sources for interpretation, verification, and auditing. Accurate business information and analysis, where every piece of data is traced back to its source and is related to its true meaning, creates competitive advantage by transparency to management. Customer-facing employees gain tremendous power to enhance their service quality, with a transparent view of a customer's personal preferences and dealings with the business.

Integration

Integration allows processes throughout the SOA to act as a single, unified process, a key driver of efficiency when communicating within the enterprise or with partners. While there are many prerequisites to achieving genuine integration, a semantically correct knowledge of IT systems (metadata) and the existence of a common business language (canonical message formats) are two of the most critical elements. Up-to-date information of IT systems and a semantic information model realized in canonical message service interfaces are two of the most critical elements in achieving sufficient integration.

For instance, a closed purchase order might require that the primary system tracking orders (e.g., an ERP system) be updated. Yet the customer relationship management (CRM), financial reporting, systems responsible for delivering the product, HR systems that track commissions, and data warehouses that track sales trends all require synchronization. Moreover, integration supports new dynamic business practices such as taking into account live sales data to automatically tune dynamic pricing or to redeploy marketing resources to maximize revenue.

Agility

Agility is achieved when semantically accurate business information ensures that service requests and responses correctly incorporate required data. When processes in SOA are modeled in terms of a common business language (instead of being tied to specific data sources), it is much easier to change business processes.

One major driver for the deployment of SOAs in the enterprise is the vision of the real-time enterprise (sometimes referred to as the "zero-latency enterprise"). The architectures are intended to allow many kinds of agility including the ability to responsively change parameters, the ability to restructure and customize business processes, the ability to prepare for organizational changes, and the ability to update, change, and integrate IT systems in response to changing business needs or new technologies.

When all business information is available enterprise-wide through a semantic information model, any field or data structure can be channeled into a service. When processes are modeled in terms of a common information model rather than tied to specific information sources or services, it is much easier to add and change processes.

Conclusion

By applying semantics, Web services break away from dependency on specific schemas and data sources. The business meaning of each service is exposed for semantic discovery. Moreover, clients can search and make calls on individual or composed services with the required functionality. Therefore, even if no one service exposes the needed interface, the client can still find the needed services, send them the required input, and transform their outputs to meet its needs.

When enterprise architects anchor their SOAs in a semantic model, integration, messaging, and other projects no longer suffer from the errors typically caused by inconsistent terminologies or definitions. As semantics boost the business value of SOA-sustaining Web services, we can look forward to a rapid growth in adoption of both semantic and service-oriented technologies.

Resources

- Borenstein, J. and , J. (2003). "Semantic Discovery for Web Services." *Web Services Journal*. SYS-CON Publications, Inc. Vol. 3, issue 4. www.sys-con.com/webservices/articleprint.cfm?id=507
- Cowles, P. (2005). "Web Service API and the Semantic Web." *Web Services Journal*. SYS-CON Publications, Inc. Vol. 5, issue 2. www.sys-con.com/story/?storyid=39631&DE=1
- Genovese, Y., Hayword, S., and Comport, J. (2004). "SOA Will Demand Re-engineering of Business Applications." Gartner. October 8.
- Linthicum, D. (2005). "When Building Your SOA…Service Descriptions Are Key." WebServices.Org. March 2005. www.webservices.org/ws/content/view/full/56944
- Schulte, R.W., Valdes, R., and Andrews, W. (2004). "SOA and Web Services Offer Little Vendor Independence." Gartner. April 8.
- W3C Web Services Architecture Working Group: www.w3.org/2002/ws/arch/

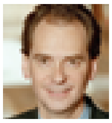
About the Authors

Joshua Fox is a senior software architect at Unicorn Solutions (www.unicorn.com), developing semantic information management systems for the enterprise. He has experience developing large-scale clustered Java systems for Internet collaboration and multimedia delivery, and has published and lectured widely.

joshua.fox@unicorn.com

Joram Borenstein is a senior marketing manager at Unicorn Solutions (www.unicorn.com), working on data semantics technologies for enterprises worldwide. His previous experience includes managing the rollout of content management software platforms. He has published and lectured extensively.

joram.borenstein@unicorn.com



Adeptia BPM Server 4.1

More than Enterprise Application Integration on steroids

Adeptia has released the latest version of its BPM server. The Adeptia BPM server is another entry in the fast-growing market for Business Process Management Tools.

WRITTEN BY
PAUL T. MAURER

People unfamiliar with Business Process Management servers can think of them as replacing the typical maintenance-heavy point-to-point systems integrations that let data flow through corporate machinery. The business process engine executes a well-defined business process and shepherds information through those complex multistep processes. I've heard BPM characterized as Enterprise Application Integration on steroids, but as Adeptia's product will show, it's really much more.

Adeptia's BPM Server lets users gradually build up processes by defining simple discrete activities. Activities are then combined into tasks that are executed by the BPM Server. The BPM server can be controlled from its browser-enabled console with one minor exception that I'll discuss later.

Activities

Activities are grouped by logical types including Sources, Targets, Schema, Data Security, Transformer, Notification, Event, Polling, Web Services, Workflow, and Native Calls.

The Source type identifies data to be read and particulars like where to find it and what transport protocol to use when loading the data.

The Target type defines the data to be written and particulars like location and transport protocol.

The schema type defines the format of the data to be read or written. All the major format definitions are supported (i.e., Text, XML, Excel, relational, etc).

The Transformer type specifies the schemas required to map data from a data source to a target.

The Notification type sends e-mail at points in a process flow.

The Event type schedules and triggers process flows.

The Polling type allows process flows to "wait" and "listen" for incoming data.

The workflow type manages user interactions with running process flows.

The Native Call type can run operating system scripts asynchronously during the execution of a Process Flow and, yes, Web Services are well supported. Overall a good set of building blocks to work with.

Process Designer

Adeptia provides a slick Process Designer (Figure 1) that lets users create process flows using BPMN Notation. Once process flows are created using BPMN graphical elements, activities are attached to the elements. This two-step process lets a business user sketch out the logical process. Later when the logical process is well understood, a technologist can locate or create all the required activities and attach them to the process sketch. This in turn creates a fully realized executable business process.

Sometimes these design tools are just eye candy. There's definitely the requisite drag-and-drop functionality for the casual user but Adeptia has done a good job of including features for the technically savvy like the ability to synchronously or asynchronously call other process flows and to peek and poke context variables. Users can control transaction contexts with Begin, End, Checkpoint, and Rollback activities. Also, adding judicious pauses and traces to a flow is a straightforward exercise.

It's interesting to note that the process designer was implemented as a Java Web Start application. This is the minor exception I mentioned earlier. The designer combines the convenience of a single license and install on the server that can do offline work on the desktop. A user can design a process flow when not connected to the BPM Server. He can switch from Offline mode to Online with the status displayed at the bottom of the process designer.

Data Transformation

Adeptia BPM Server can transform one type of data into another type facilitating any-to-any transformation. The data can be transformed in one of three ways:

- Using the Data Mapper
- Using a Record-to-Record Service
- Using a Scripted Service

The Data Mapper (Figure 2) is a visual mapping tool that lets users specify Source and Target Schemas selecting data fields to map between the two. A Map can be of the simple field-to-field variety, but can also include mapping functions that allow string and Boolean manipulation, math functions, and aggregation. The Adeptia mapper includes a nifty "auto-mapper" feature that attempts to match fields between source and target schema automatically. The auto-mapper is smart enough to account for case and attribute sensitivity as well as leaf-count matching. This definitely

FREE RESOURCE CD INCLUDED!
information
STORAGE+SECURITY
journal
www.ISSJournal.com
In this issue:
IP Storage
SAN NAS
Disaster Recovery
Enterprise Security
Infosecurity
PREMIER 2004 VOL 1 ISSUE 1
Microsoft RSA Security
SecureID for Windows
SAN NAS
Emerging technology trends and market maneuvers

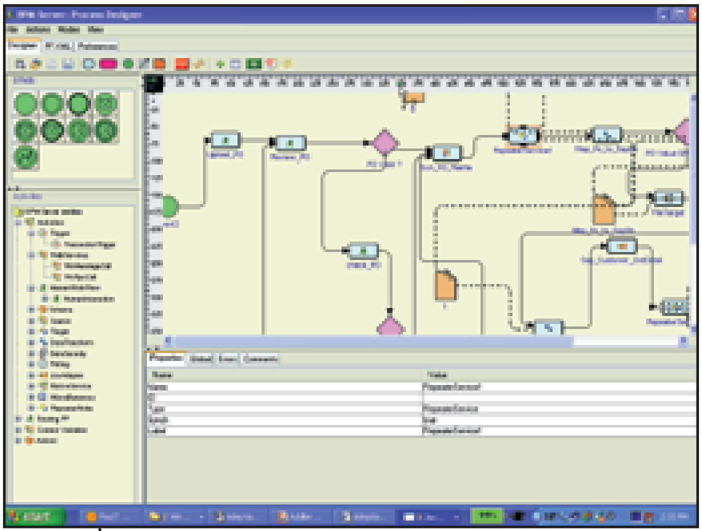


FIGURE 1 Adeptia Process Designer

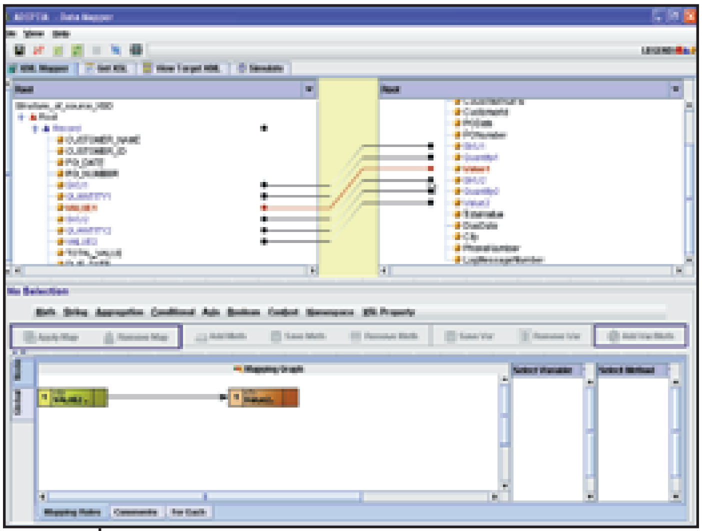


FIGURE 2 Adeptia Mapper

saves time for the more mundane, yet verbose schema.

Where Adeptia steps out ahead of other-mapping tools I've worked with is in its support for testing a finished mapping. The mapper generates XSL that can be viewed in the mapper's "Get XSL" tab. There are tools in the mapper for validating the XSL, generating mock input data, and stepping through the XSL visually as it runs. Nice!

Reporting

Now that we've talked a bit about creating process flows and mappings, how do we know that they're running or have run?

Well, Adeptia includes browser access to its system logs and there's a wealth of information to be found there, but when you need to hone in on a process and find out what's going on, the process flow log is where to look. The browser interface to the process log lets users search for a particular flow by name, status, and date range. The report returns the runtime, name, status, and completion message for all matched flows. Clicking on the "view details" link for a flow displays the activities that ran during the process execution including timestamp, activity name, type, status, and completion message.

Adeptia also gives users the ability to build custom reports using Jasper templates. Jasper is an open source Java reporting tool that lets users design custom report templates in the XML format. Note that Jasper isn't provided with Adeptia's BPM Server.

Reliability

Reliability is always a concern in the enterprise and Adeptia covers its bases by providing clustering capabilities. Its clustering supports load balancing, which is used to distribute the execution of process flows evenly between the members of a cluster. Load balancing is supported using a round-robin algorithm. The clustering also provides failover capabilities. If a node in the cluster falters another node will pick up and finish the process.

Industry Templates

Adeptia also includes a number of pre-built industry-specific templates. Assets like data formats and sample process flows for standards such as ACORD, EDI RosettaNet, AIAG/APQP, etc. With these kinds of tools, preloaded templates can give a project team a leg up on the work.

Documentation

Adeptia provides online help as well as a User Manual that's just shy of 400 pages.

Conclusion

Adeptia BPM Server 4.1 combines Business Process Management capabilities with strong integration capabilities. Its BPMN standards-based process designer should feel natural to BPM experienced business people but still lets the technology team polish things to make a process robust. If you're dipping your toe in the waters of BPM, you should check out Adeptia's BPM Server product. ⓔ

About the Author

Paul T. Maurer is a principal in the financial services practice of a leading consulting services company.

■ ■ ■ paul@paulmaurer.net

ADEPTIA

Company Info
Adeptia

20 North Wacker Drive
Suite 2000
Chicago, IL 60606

Phone: (312) 442-4420
Fax: (312) 263-0636

Web: www.adeptia.com

Licensing Information
BPM Server 4.1 Express: free – Limited to one-user two-process flows
BPM Server 4.1 Professional: $40, 000
BPM Server 4.1 Premiere: $80, 000

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

REST vs. SOAP: The Battle of the Web Service Titans

Picking a winning Web service approach

■ At their most basic level, Web services work well and solve important problems because they are based on standards. However choosing among the available Web service approaches can still be surprisingly difficult. Whether it is correct or incorrect, the choice will have far reaching consequences for any project. A foray in the wrong direction often means having to abandon the initial investment, in terms of development, training, and infrastructure, to retool everything to go another way. Make a good decision though, and Web services become an enabling springboard to enterprise integration, information self-service, and all the attendant benefits of service-oriented architecture. In a highly competitive business environment, the risk of lost time and investment means that making the best decision up front is vital to success.

WRITTEN BY
DION
HINCHCLIFFE

Unlike a few years ago, there are now many good ways to build standards-based Web services. Some are easier and some are certainly better than others, but for most projects, it generally boils down to deciding between two competing technical approaches: SOAP or REST. Since they both take the Web service creator down uniquely different paths, it's worth spending time determining exactly where each approach leads. Even with care, making the decision correctly is difficult and will certainly be definitive for a Web services development effort. The difficulty is compounded by the ongoing debate over whether SOAP or REST is best for a particular application. Certainly there will be groups within an organization championing one direction or the other, but fortunately for those making decisions today, there is now hard-won field experience and lessons learned about both approaches to draw from. It's time to take a close look at where SOAP and REST are, what they are really capable of, and where they are going.

Though SOAP and REST are essentially the same in what they do, they couldn't be more different in terms of how they work. Explained simply, SOAP and REST allow programs to exchange XML messages over a network, usually some flavor of the HTTP protocol used to serve up Web pages. HTTP is the ubiquitous workhorse protocol that forms the infrastructure of the Web and is used by Web clients billions of times a day to send and retrieve pages from Web servers around the world. By leveraging the proven and well-supported HTTP protocol for the physical transmission of pure data instead of visual Web pages, SOAP and REST provide clients access to reusable information services over a network while using technology standards that IT departments are already intimately familiar with. It's no accident that because they both use popular and widely supported standards to provide services, SOAP and REST are held up as poster children of the Web services revolution. Web services are about interoperability, and SOAP and REST deliver the goods.

First described by Roy Fielding in his PhD dissertation, Representational State Transfer (or REST) is based on the concept of transferring state between two systems on a network. This method is identical to the way in which a Web browser gets Web pages from or posts them to a Web server. With REST, the pieces of data that are trans-

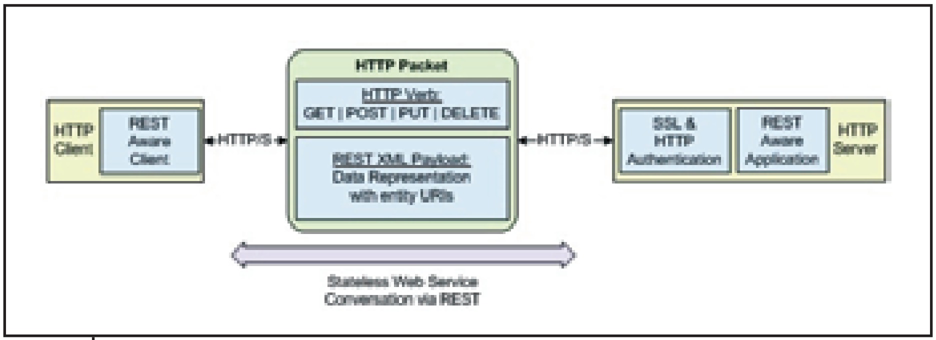


FIGURE 1 Conceptual diagram of REST

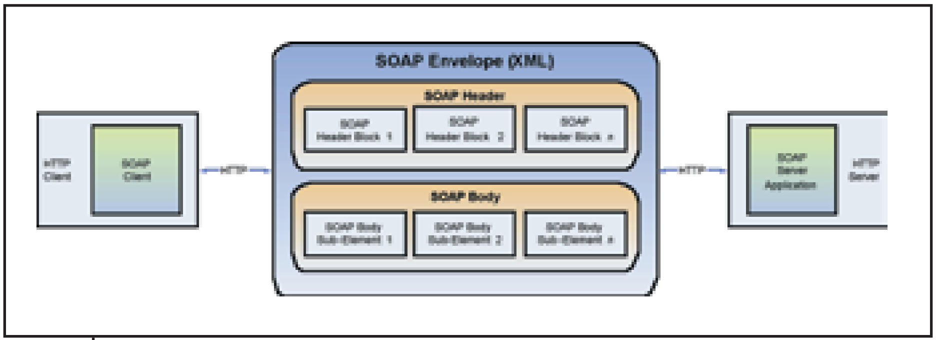


FIGURE 2 Conceptual diagram of SOAP Web

ferred by the Web service must be identified individually using a standard addressing scheme. REST developers are encouraged to think of these addresses in terms of something called a Universal Resource Identifier, or URI, of which the ubiquitous Web page URL is subset. A URI uniquely identifies any kind of network resource for which a unique address can be created. In practice, URIs are assigned to individual data records, such as customer records or orders, transferred between the Web service and its client.

REST's URL-style lookup model is eminently familiar to all Web users and is just one way that REST leverages the successful aspects of existing Internet protocols. To cap off the REST model, the URI and an HTTP verb (GET, POST, PUT, DELETE) are used together to issue a request to a REST Web service to access and manipulate the server-side state, or data record, identified by the URI (see Figure 1). The actual XML message is contained in the HTTP request and security is provided by HTTPS, which is the secure version of HTTP. This, in a nutshell, is virtually everything that a Web service user or creator needs to know about REST.

In the other Web service camp is SOAP, a more formal, deliberate, and standardized solution to providing Web services. Fortunately, the structure and internal moving parts of SOAP messages are generally well organized and straightforward. Like REST, SOAP is an open messaging framework based on XML but is backed by formal standards from a recognized standards body, the World Wide Web Consortium (W3C). SOAP also has compelling built in extension mechanisms that make it an excellent choice as the basis of a Web service stack on which to build more advanced capabilities.

Revised several times since it was originally issued in 2000, the current version, SOAP 1.2, now lets the Web services creator decide on the exact network protocol to use. This can still be HTTP but could just as easily be SMTP, the Internet mail protocol. SOAP messages themselves are embodied by an XML document known formally as the SOAP envelope.

The SOAP envelope consists of two parts, a header and a body. The header contains optional control information about the message. The SOAP header is both novel and highly useful in that it allows SOAP messages to be annotated in a standardized way (see Figure 2). This allows a SOAP message to carry along descriptive metadata information about security, transactions, or other useful context that does not change the message itself but describes higher-level processes of which it is a part. The SOAP body portion of the envelope contains the actual XML message being transmitted.

Appreciating the standardized yet extensible design of SOAP is part of understanding the unique appeal of the SOAP model. Because many other interesting Web protocols can and do plug into SOAP via its header mechanism while treating the message carried as inviolate, SOAP becomes a powerful intermediary between sophisticated systems that can process various parts of the message. These services can plug into SOAP messages by providing extensions for security, orchestration, transactions, and other features. It is specifically this capacity for extension that makes SOAP so interesting to organizations wishing for standardized ways of using sophisticated, enterprise-class capabilities with their Web services. If nothing else, SOAP will remain popular for this capability, though other features, such as having schema support, also make SOAP a good choice for its ability to describe the data it carries.

Beyond the Basics: SOAP/REST Differentiators

Beyond their similar means for XML message exchange over HTTP, SOAP and REST have little in common. Unlike REST, SOAP is a recognized W3C standard that describes a consistent way to exchange XML messages. In SOAP-based Web services, XML formatted SOAP envelopes are exchanged over a selected protocol with XML messages that are described with a schema. The wire protocol is flexible with SOAP, which it isn't with REST, and SOAP is able to use a growing variety of other network protocols in a standardized fashion. SOAP's message format is more technically intricate and rule-constrained than REST, but this also facilitates the underpinning of a layer of associated Web services that are based on a series of other standards. This Web services stack ostensibly offers a "solved" Web services architecture to organizations that want it. It's worth noting though that many of these value-add Web services are still in development, well-known projects such as

REST vs. SOAP Scorecard

Issue	REST	SOAP
Standards-based	Yes, existing standards like XML and HTTP	Yes, old and new standards together
Development tools	Few, not largely necessary	Yes, plentiful commercial and open source
Management tools	Uses existing network tools	Yes, often costly but in abundance
Extensible	Not in a standards-based way	Yes, many extensions, including the WS-* standards
Easy to implement	Yes	Yes, but only if you have a SOAP-enabled environment
Training and support resources	Limited	Yes
Transaction-aware	Not automatically, must be supplied by user	Several standards-based solutions are available
Service-oriented architecture friendly	Limited, few building blocks for advanced service-orientation exist	Yes
Platform restrictive	No, easy to use from virtually all OS, language, and tool platforms	Somewhat, the more of SOAP used, the more restrictive it can be
High performance	Yes	Slower than REST, but SOAP/1.2's binary compression may change that

Microsoft's Indigo project, which isn't due for many months, are intent on delivering advanced services on top of SOAP for widespread use.

However all complexity comes at a cost. This means SOAP can be difficult to deal with when things go wrong and the problems have to be figured out. To allay this somewhat, SOAP also has quite good tool support, with many commercial software vendors offering SOAP-aware products across the spectrum of design, development, monitoring, and management. In contrast, software support for REST seems relatively stark, especially on the management and administration side. Many REST advocates would counter that existing tools that manage HTTP and SSL already do the job.

SOAP's formalized and federation-friendly nature stands in almost direct opposition to REST. In reality, REST isn't even a Web services standard. It is an architectural style that leverages the popular Internet standards XML and HTTP. REST is about, and really only about, the XML message being exchanged over a network. To understand REST, one merely needs to visualize a block of XML and think about HTTP's simple verbs (GET, POST, PUT, and DELETE). REST uses these verbs to access and manipulate the information like a resource on the network, like a browser uses a Web page. REST can appear simplistic, even crude, when compared to SOAP, but in reality, most users find it lightweight, flexible, and easy to use and debug.

Due largely to its lack of complexity, relative ease of development, and little need for tools, REST has been extremely successful in grabbing development mindshare. REST also has a proven track record for performance and scalability. For example, Amazon reports that it now has over 50,000 users of its Web services, and most of them are using the REST interface. Fewer than 20% of Amazon's Web service clients use Amazon's SOAP interfaces. Also, while Amazon reports SOAP usage is trending upward, REST usage continues to climb in other places, including GM's massive internal Web services restructuring.

From a tool perspective, REST, unlike SOAP, has comparatively little commercial software support, although this is often considered a feature because REST doesn't actually need much in the way of tools that aren't already at hand, such as a Web browser for debugging.

Summary of SOAP and REST

REST

- Messages are represented in plain XML
- HTTP is used for the transfer protocol
- HTTP verbs are used for access/manipulation commands
- URIs are used to uniquely identify resources in message
- HTTP authentication provides security
- There is no formal method for expressing the interface contract

SOAP

- Messages are represented in a standardized XML SOAP "envelope"
- Can be bound to various protocols including HTTP and SMTP
- Access to and manipulation of data are application specific
- Security is not described by SOAP and is to be provided by the developer
- XML schemas are used to define the contract between client and service

In the past couple of years there has been intense debate within the Web services community about the virtues of both SOAP and REST, or their perceived lack thereof. Despite almost universal support by the .NET and Java platforms, there are many who believe that SOAP is having some adoption challenges in the industry, mainly due to perceived complexity. On the other hand, REST has been quietly growing in popularity, partially as an underdog and partially because it is not exclusionary, but mostly because of ease of implementation. At the same time, supporters of SOAP are continuing to make it more robust, more extensible, and easier to use.

So which approach is best? It really depends on the requirements, but there are certainly compelling reasons for using either. REST thrives on its spare design and relatively informal techniques and it requires no additional development tools, yet SOAP is powerful and sophisticated (though sometimes to the point of obscurity). The development and management tools available today for SOAP matured considerably and their features are compelling. For many applications, especially full-blown service-oriented architectures, the various W3C Web service additions (known as WS-* extensions) are almost a necessity.

When it comes down to meaningful differentiators, an important difference between REST and SOAP is the ease with which management and troubleshooting can be done with them. SOAP is very routing- and processing-centric, with various flags that indicate whether a message must be fully understood by an intermediary and dealt with. For its part, REST can be debugged with a garden variety Web browser. As for true interoperability, REST message aren't standardized, which can make them harder to manage or monitor using off-the-shelf tools since their content is not documented in a way that commercial tools can understand. SOAP however has an extensive array of professional applications that can be used to manage, monitor, route, and process it.

Due to lower messaging overhead, REST is generally considered faster than SOAP. Though use of text-based XML in Web services already causes bloat on the network with both approaches, SOAP, which is often used in conjunction with the WS-* protocol extensions, further exacerbates the problem by adding special handling and processing requirements. Amazon, one of the leading implementers of the REST, claims that REST is up to six times faster.

Another important way of looking at Web services approaches is to consider them from an interoperability standpoint. Proprietary products like Web service development tools, platform-driven Web services like .NET's .asmx approach, and legacy system integration efforts that expose older systems through Web services typically use SOAP instead of REST. This tends to pollute the Web service with idioms and patterns from the underlying system, language, or tool. This makes it much more difficult for those without the same environment to consume them. But those with the same Web service toolkits on both ends of the network will likely have much less trouble. They will also be able to take advantage of more advanced layers of services like those built on top of SOAP.

Fundamentally, Web services are about the XML message itself, the angle brackets and text that travel on the network wire. The contract that a Web service offers, namely the promise to do something in exchange for communicating using an agreed upon message format, is the key to the promise of true interoperability between communicating systems of Web services. The use of plain XML as the native data format of Web services means that they are usable from any client, any system, anywhere in the world. Anything that convolutes this process reduces the ability to interoperate. Major organizations with international presence, such as Amazon and Flickr, have proven that Web services can be consumed successfully by a large and highly diverse audience. REST tends to support and embrace this diversity and foster fundamental interoperability, albeit in a different way the SOAP does, which uses large, formal standards to try to achieve the same result.

In the end there is no clear winner for every application. Selecting a Web service approach is highly situation based. REST is a very attractive choice for more basic applications that involve high levels of interoperability between multiple platforms. SOAP is very appropriate for larger, formal applications that require advanced capabilities between relatively homogenous systems. SOAP's ability to leverage standard Web protocols such as WS-Security, WS-Policy, and WS-Transactions make rich functionality easy to deliver quickly, but at the expense of interoperability.

Unfortunately, users without ready access to working implementations of W3C Web service standards in their development toolkits are unlikely to be able to develop support for them on demand. This means SOAP's ability to layer advanced capabilities together, perhaps too easily, tends to shut out those that don't have ready access to them. Some observers take the long view, that SOAP, and its associated WS-* standards will ultimately be ubiquitous and no one will lack the ability to interoperate. Adherents to REST will be achieving value along the way, asking all along why it took so long for everyone to get there.

In the end, however, the details of the Web service wire protocol are growing increasingly less interesting as the real challenges become clearer. The power and appeal of Web services are often lost when the focus is exclusively on the lowest level of the Web services story. There is also a growing concern in the industry that the proliferating array of Web service standards and efforts is fragmenting the market and decreasing their ability to interoperate.

Beyond this, though, Web services deliver far more value when the view goes beyond the simple messaging that SOAP and REST provide. Though this decision on the Web service approach is important, the real goal is to get beyond this point, to the level of service-oriented architecture, where more meaningful concerns become visible. Issues such as information policy, business processes, governance, and consistent semantics are what make the Web services interesting, and more important, what makes them a genuine contributor of value to the organization.

About the Author

Dion Hinchcliffe is senior architect at Sphere of Influence with 15 years of enterprise architecture experience with various Fortune 1000 customers. Dion consults, thinks, and writes about enterprise architecture, service orientation, and agile processes, and is located in Washington, DC.

■ ■ ■ dhinchcliffe@sphereofinfluence.com

Understanding Coupling in the Context of an SOA

Promoting protocol-based integration

As we bring our SOAs online using Web Services, we all know that SOAP is the standards message transfer protocol. But the interface description language for Web Services (WSDL) isn't specifically for SOAP. It's more generic. A SOAP-centric contract description language for Web Services is needed. Enter the SOAP Service Description Language or SSDL. This framework was just released in mid-February, and is now available for review. Let's take a look.

WRITTEN BY
DAVID LINTHICUM

SSDL lets developers publish and share descriptions of the messages and message exchange patterns supported by Web Services. It also addresses contract and protocol description. In other words, it provides a framework for managing common abstractions such as protocols, messages, and even endpoints. An SSDL contract also acts as a placeholder that encapsulates protocol descriptions, as well as a mechanism for those protocol descriptions to reference the declared messages using a common facility. The frameworks are built very much like the WS-Policy specifications, using a base SSDL framework that offers a fundamental protocol for building and describing messages.

Under Contract

The SSDL provides four protocol description frameworks, but others are possible down the road because it's extensible. The initial release includes: Message Exchange Patterns (MEP) Framework, Communicating Sequential Processes (CSP), Rules-based SSDL Protocol framework (Rules), and the Sequencing Constraints (SC) Protocol Framework.

MEP defines a collection of XML infoset elements information representing commonly used exchange patterns. *CSP* defines a collection of XML infoset element information items for defining a multi-message exchange, leveraging sequential process semantics. *Rules* defines a collection of XML infoset elements information used to describe a multi-message exchange protocol using conditions. *SC* defines a collection of XML infoset element information for multi-party, multi-message exchange protocol using pi-calculus notions.

The goal here is to support various types of end users, letting them pick and choose protocol frameworks that match their applications best. Or, as an option, they can implement their own protocol framework (see below).

SSDL assumes that SOAP is the transport mechanism. So, you can consider SSDL bound to SOAP. SSDL also assumes that WS-Addressing is the standard mechanism for embedding addressing information inside of SOAP envelopes. SSDL focuses on messages and protocols, and as a result there's no need for articles such as 'interface,' 'inheritance,' and 'operation.' An XML infoset is assumed to be the SSDL component model, and the modularization of the contract is handled using XInclude.

As mentioned above, SSDL promotes protocol framework extensibility, allowing a different protocol description model to be integrated into the base SSDL framework, promoting protocol-based integration and letting you expose the message behavior of a service. It is key to the success of the standard, letting flexibility support protocol integration, something that's been difficult in the past.

What's the Value Prop?

So, what do we get here?

SSDL is really providing the mechanisms for service architects to describe the structure of SOAP messages. Once a message has been described, one of the currently available protocol frameworks can be leveraged to combine the messages into protocols that expose the message behavior of the service. As such, architects and developers can create systems that participate in conversations, all adhering to a contract.

The SSDL contract lets those that design and build services focus on the structure of the messages, as well as the message exchange patterns, again by using a common contract for designing, building, and deploying a service. In the SSDL world there are no abstract interfaces, inheritance, or operations. The building services using SSDL define protocols by correlating messages together. That's a powerful concept, and an approach that has obvious advantages. So, how do you play with this thing now?

A tool called SSDL.EXE exists created using the .NET 2.0 platform and Web Service Enhancement 2.0 to consume SSDL contract. SSDL.EXE generates C# and VB.NET code that can be leveraged to implement Web Services for producing and consuming messages. It's extensible through plug-ins, and is only Win32-based so far.

I think we need something like SSDL as SOAs become the rule rather than the exception, providing better control and definition of information moving from point to point. This standard seems to work and play well with other standards, but I think it's a bit complex for many architects to understand. Its extensibility is very powerful, and I suspect will be leveraged in many SOA development projects. Keep an eye on this one.

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. You can reach David at dlinthicum@grandcentral.com.

■ ■ ■ dlinthicum@grandcentral.com

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

Pick a 3-Pack, a 6-Pack or a 9-Pack

□ 3-Pack	□ 1YR □ 2YR	□ U.S. □ Can/Mex □ Intl.
□ 6-Pack	□ 1YR □ 2YR	□ U.S. □ Can/Mex □ Intl.
□ 9-Pack	□ 1YR □ 2YR	□ U.S. □ Can/Mex □ Intl.

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

□ **LinuxWorld Magazine**

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **JDJ**

U.S. - Two Years (24) Cover: $144	You Pay: $99.99 /	Save: $45 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $89.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Information Storage + Security Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $39
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ **Wireless Business & Technology**

U.S. - Two Years (12) Cover: $120	You Pay: $49.00 /	Save: $71 + FREE $198 CD
U.S. - One Year (6) Cover: $60	You Pay: $29.99 /	Save: $30
Can/Mex - Two Years (12) $120	You Pay: $69.99 /	Save: $51 + FREE $198 CD
Can/Mex - One Year (6) $60	You Pay: $49.99 /	Save: $10
Int'l - Two Years (12) $120	You Pay: $99.99 /	Save: $20 + FREE $198 CD
Int'l - One Year (6) $72	You Pay: $69.99 /	Save: $2

□ **MX Developer's Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ **WebSphere Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **WLDJ**

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
Can/Mex - Two Year (12) $120	You Pay: $69.99 /	Save: $50
Int'l - Four Years (24) $240	You Pay: $120 /	Save: $120 + FREE $198 CD
Int'l - Two Year (12) $120	You Pay: $79.99 /	Save: $40

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

3-Pack

Pick any 3 of our magazines and save up to $210

Pay only $99 for a 1 year subscription plus a FREE CD

- 2 Year – $179.00
- Canada/Mexico – $189.00
- International – $199.00

6-Pack

Pick any 6 of our magazines and save up to $340

Pay only $199 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $379.00
- Canada/Mexico – $399.00
- International – $449.00

9-Pack

Pick 9 of our magazines and save up to $270

Pay only $399 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $699.00
- Canada/Mexico – $749.00
- International – $849.00

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA

XML JOURNAL SUPPLEMENT

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

This Month

XML Schema May Not Always Be the Right Tool for the Job

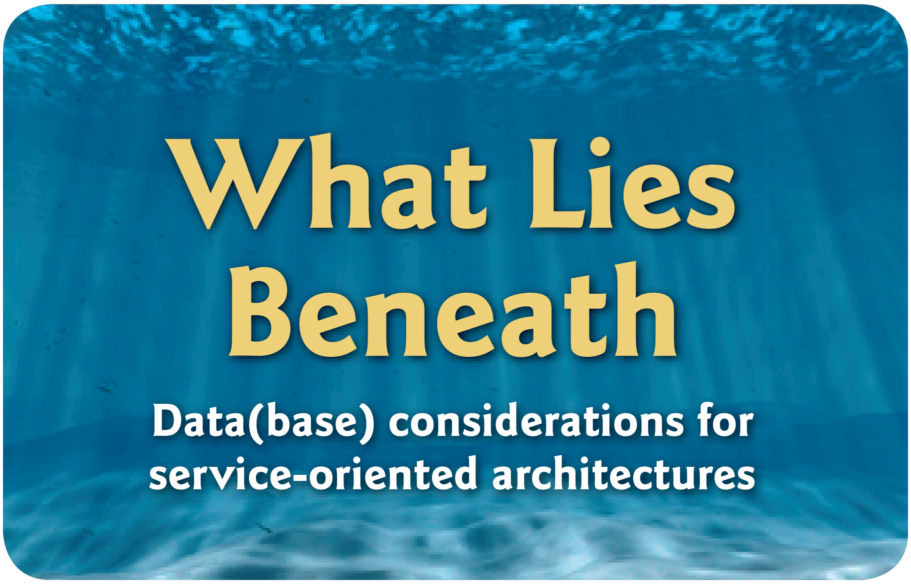
Michael Wacey

Will many of the features in XML Schema be widely used? In particular, I agree that it is better to have an XML language for specifying document layout rather than the DTD language. On the other hand, I am not sure that the document layout should be strongly typed. The nightmare scenario is where a customer cannot place a large order because an XML document is invalid. Assume a company has an average order size of $50,000 dollars with a current order range of $3,000 to $220,000. It would be reasonable to set a criterion in the Purchase Order Schema for this company to set a total order rage of Zero to One Million.

Overview and Analysis of the Department of Defense Discovery Metadata Specification (DDMS)

Michael Sick

The Department of Defense (DoD) Discovery Metadata Specification (DDMS) describes the DOD's preferred approach for decorating data assets with metadata. By providing a common convention for metadata, the DoD is building a common system for asset discovery, search, description, consumption, and security. This article provides a summary of the DDMS's purpose, structure, and capability. Upon completion the reader should have a basic understanding of the DDMS and should know where to go to get more detail and related materials. All questions regarding this article should be directed to Michael Sick at mike@serenesoftware.com.



What Lies Beneath

Data(base) considerations for service-oriented architectures

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

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

What Lies Beneath

Data(base) considerations for service-oriented architectures

WRITTEN BY
Rajan Chandras

How important are data considerations to service-oriented architectures (SOA) vending Web services? Consider the following definition of Web services from AMR Research: *"Web services have been commonly defined as a standardized way of integrating applications using Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Definition Language (WSDL), and Universal Description, Discovery and Integration (UDDI) standards. XML is used to <u>tag the data</u>; SOAP is used to <u>transfer the data</u>; WSDL is used for <u>describing the services</u> available; and UDDI is used for <u>listing what services</u> are available."* (All the emphasis is mine).

In other words, we use technologies such as UDDI, WSDL, SOAP, and XML to find, understand, package, and format Web services – but the services themselves are all about managing data and providing data-based behavior. Now, this is neither the time nor the place to get into a "process v/s data" war. The point is simply that in order for service-oriented architecture to deliver up to expectations, there must be serious consideration of data-related factors. What are these factors? From the perspective of data management, interestingly enough, these factors are essentially no different for Web services architecture than for other application architectures, such as a client/server application or a conventional Web-based transaction processing application. (Conventional and Web-based together? How the times have changed.) A note to the purists: for the purposes of this article, "service-oriented architecture" and "Web services" are used interchangeably – in truth, of course, any loosely coupled, layered architecture where one or more layers exposes its functionality in the form of callable services may be deemed as a "service-oriented architecture."

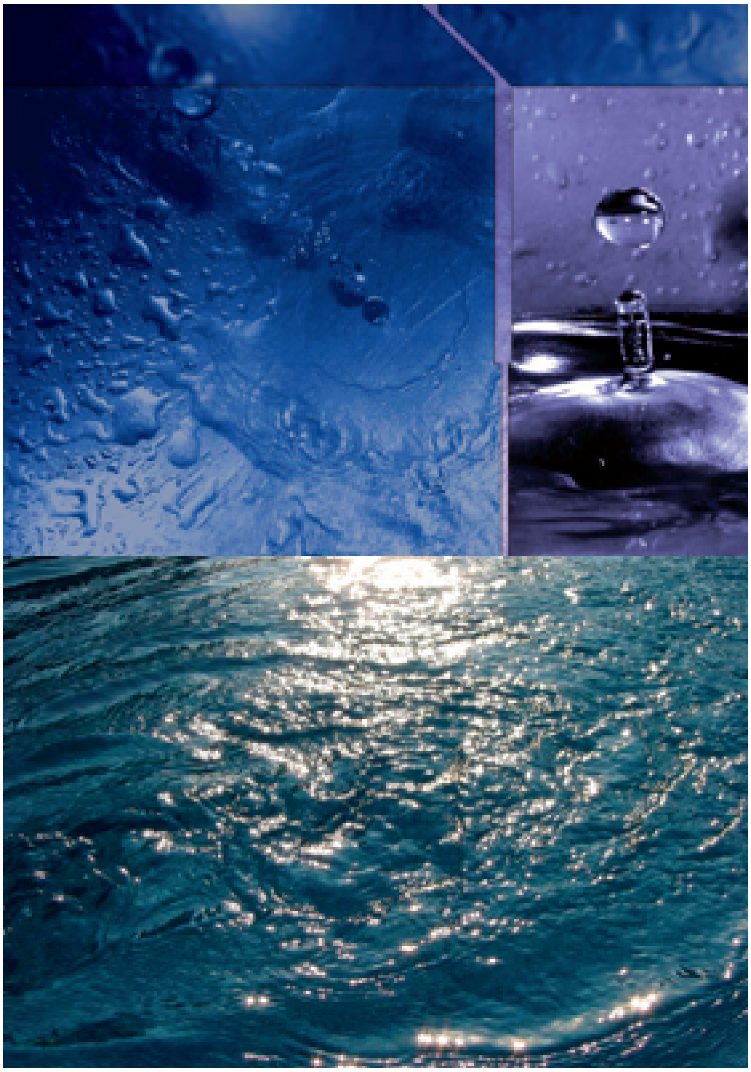
Data Integration

Perhaps the most important aspect of data management is the integration of various components and layers of data. The greatest threat to the success of SOA lies here: fail to integrate all your data, and the application may tend to get increasingly *dis*service-oriented. As our information systems move from being soloists to joining ensembles, chorus and symphony are the name of the game. It is no longer sufficient that an application or a component of a large architecture manages its data correctly – though let's face it, even this is often no ordinary task. In addition, the component data must stay coordinated and "march in step" with other components. In a research paper on IT Trends 2004, Philip Russom of Giga Research (a wholly owned subsidiary of Forrester Research), identified the following key trends in data integration. In the paper, Mr. Russom discusses "data integration" from a more classical and thus somewhat different perspective than I have in mind; hence, although the bullet titles are borrowed from the paper, the ordering, interpretation, and corresponding descriptions are my own.

> **"Perhaps the most important aspect of data management is the integration of various components and layers of data"**

- ***Convergence between data integration and application integration.*** No longer can data integration be viewed as a concern distinct from (and independent of) application integration. If we consider Web services architecture as a collection of semi-independent applications, then integrating these applications into a single cohesive architecture must pay attention to how the corresponding data repositories integrate, not just at the physical (database) level, but more important at the conceptual and transactional level.

- ***Distributed architectures for data integration.*** Today, traditional (or classical) data integration has come to connote the bringing together of widely disparate and distributed data by various means such as data extraction-transformation-loading (ETL) and enterprise information integration (EII), which requires a distributed architectural approach (and curiously enough, may also use Web services!). On the other hand, while SOA may not quite be "distributed" in nature (or at least by definition), it is certainly not a conventional, centralized architecture: it falls, as a loosely coupled architecture, somewhere in the middle. From a data integration perspective, this loose coupling leads to considerations that are very similar to those in distributed data architectures: we would still have to deal with disparate data sources and data formats, and data integration is still *de rigueur*.
- ***Real-time data integration.*** Perhaps the greatest data management challenge for service-oriented architectures is real-time data integration. Whereas conventional data integration – such as ETL – often contends with (and is content with) data integration with a degree of latency, the on-demand nature of Web services requires an on-demand integration of data. It is important to understand, however, that "on-demand" is not quite the same as "on-the-fly." On-demand (or real-time) data integration typically requires significant upfront planning, design, and development efforts in order for the promise to be realized. Once again, think symphony.
- ***Virtual and federal data integration.*** If the problem is distributed data architectures and the need for real-time data integration, then consider using data virtualization techniques in your SOA. A "required competency" for service-oriented architectures that is reborn in the forceful new *avatar* of Enterprise Information Integration (EII), data virtualization is now available in the form of sophisticated, off-the-shelf solutions that you more or less plug into your SOA. Of course, this does not imply that the need for a planned approach for data integration is diminished in any way – in fact, quite the contrary.
- ***Extreme data integration scalability.*** In Web services more than anywhere else, data volumes are a threat to data integration. As data volumes grow, data integration for Web services is exposed to performance and scalability challenges that can derail the best architected SOA. As the number of "moving parts" increases (both in terms of number of components and the amount of data brought in and exchanged by the components), the architecture may begin to exhibit signs of stress that were previously undetected. This is the second-greatest challenge to SOA from a data management perspective.
- ***Evolving economics of data integration.*** Data integration can be an expensive proposition in itself. Mr. Russom notes that one of the curious epiphanies striking users as they study the economics of data integration is that the cost of hand-coding is misleadingly low in the first phase of a project, but goes through the roof in maintenance phases, and this realization has accelerated the trend away from hand-coding and toward solutions developed atop a vendor's platform. This is true enough, yet, on the other hand, vendor-provided ETL tools and EII and Data Quality solutions don't come cheap either – nor do consulting services to deploy these in the enterprise and leverage them in order to gain a meaningful return on investment (ROI). In the context of Web services, architects must pay careful attention to their data integration needs, and select only those tools and technologies that deliver the need – which, in turn, implies that the need for data integration must be assessed first, followed by careful planning and design.



Data Quality

Next in importance to data integration (or perhaps more important than it) is the need for data quality. Quite simply, data quality will make or break the service-oriented architecture. Data quality is a fairly wide discipline, but there are two aspects to data quality that are most pertinent to SOA: structural data integrity and data consistency. Structural data integrity comes into force at the data modeling level, and ensures that all database objects and components are structurally sound; data consistency comes into play when the transactions get going. Examples of structural data integrity and data consistency are:

- Do all tables have primary keys? (I have seen this one violated more often than you might imagine.)
- Do common attributes share a common definition? (E.g., is the Customer ID defined identically in various places?)
- How will referential integrity be maintained across component databases?
- How is reference (or lookup) data managed across databases?
- Is there a consistent approach to maintaining historical data?

As you might imagine, unresolved issues of these kinds can throw a sizable spanner in the SOA machinery.

> "Any discussion on data today would be incomplete without examining the security aspect of information"

Data Security

Any discussion on data today would be incomplete without examining the security aspect of information. It would be imprudent to assume that the authentication and authorization mechanisms in the SOA are fully sufficient to ensure data security, although they are clearly an important component of data security. In another insightful paper from Giga Research, Noel Yuhanna recommends a comprehensive DBMS (database management system) security architecture, comprising the following items. Once again, I have rearranged the order for the present purpose, although this time the original descriptions are retained.

- ***Application-level security.*** When application or Web servers are deployed, end users should not have direct access to the DBMS but only through the applications. Remember to apply standard practices, such as (1) allowing users one login to an application, (2) discouraging shared logins, (3) performing regular audit checks to track suspicious activities, and (4) creating a password aging policy.
- ***Data protection.*** Enterprises that store sensitive data including credit card information, social security number, and other personnel-related information should encrypt the underlying data to enable an additional security layer. However, only encrypt a few columns to minimize system overhead and performance implications.
- ***DBMS hardening.*** Most enterprises typically use the default database network port address on their database server, which often leaves the database vulnerable. Avoid using such default port numbers by assigning a unique network port address for each database application. When applicable, apply DBMS and server-level security patches to minimize known vulnerabilities; however, always perform end-to-end testing and integrated testing of the application to ensure that the security patches do not impact any application's functionality.
- ***Secure administration.*** Protect backup tapes and ensure that unauthorized personnel do not have access. Restrict physical access to the database server at all times. For test databases, use old archived data or test data instead of production data to minimize developers' and testers' unprivileged data access.

Business Continuity

In the context of service-oriented architecture, business continuity from a data perspective presents a unique challenge. Remember that a typical SOA is likely to have several databases merely to accommodate the services infrastructure: these would not include databases holding transaction (or business) data. Thus, we now must confront two levels of business continuity of data: at the business level (i.e., for transaction data), and at the infrastructure level (i.e., for data related to components of the SOA). Of these, the former is easier to handle: what is more complex, perhaps, is ensuring coherence and continuity of component data. How do we ensure, for example, that our authentication, authorization, and provisioning databases are consistent, not just during live operations, but also were we to recover from system failure? How do we arrange the seamless introduction of a new infrastructure component into the complex data interactions of existing SOA components?

Conclusion

Clearly, service-oriented architecture is about more than simply pulling together various off-the-shelf components and using these to publish services. It must be understood that the glue that holds together all of these components is the underlying data used by each component. Pay attention to this data, and half your battle is won. How do we do that? Well, in short, as I once read on a highway: "Start Early. Drive Cautiously. Reach Safely." Do you agree? Do you feel differently? I invite you to share your experiences in data management for Web services.

Author Bio

Rajan Chandras is a principal consultant with the New York offices of CSC Consulting (www.csc.com) and can be reached at rchandra@csc.com. He has written this article on his own. The author would like to acknowledge research quoted from AMR Research and Forrester Research.

RCHANDRA@csc.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

WebServices 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.

WRITTEN BY **MICHAEL WACEY**

XML Schema May Not Always Be the Right Tool for the Job

Some things to consider as you put XML Schema to use

Will many of the features in XML Schema be widely used? In particular, I agree that it is better to have an XML language for specifying document layout rather than the DTD language. On the other hand, I am not sure that the document layout should be strongly typed. The nightmare scenario is where a customer cannot place a large order because an XML document is invalid. Assume a company has an average order size of $50,000 dollars with a current order range of $3,000 to $220,000. It would be reasonable to set a criterion in the Purchase Order Schema for this company to set a total order rage of Zero to One Million. At some point in the future, after price increases and business changes, a large customer may place an order for $1,020,000. The Purchase Order document would be rejected as invalid.

Having been involved with Markup since the middle 1980s, I have seen it go through many changes and variations, from GML to SGML to XML. Many specific markup languages such as HTML have been developed. During this time, there has always been a balancing act of content definition vs. content presentation. Most but not all markup languages were defined for presentations purposes. HTML is one of the most widely deployed presentation languages.

Content-only languages are rarer and more specialized. For example, the XML-based languages defined by the RosettaNet effort and many other eCommerce activities are focused on content. In these cases, domain-specific knowledge tends to influence the definition of the markup. For example, a purchase order for the Electronics Parts industry would include information about packaging, environmental limits, and electrical specifications.

> "The nightmare scenario is where a customer cannot place a large order because an XML document is invalid"

On the other hand, a purchase order for the Bulk Chemical industry might include information about purity levels, manufacturing process, and material safety.

It is natural to desire to place validation criteria along with the structure of the documents. For example, if there is a field called Date Required, you would want to make sure it is a valid date. This will ensure that only documents with valid dates are accepted. This leads to document definitions that are very well constrained. From a receiving viewpoint, this is great because it makes sure that all documents are well defined and well organized.

The picture from the sending side is more complex. The sending company may not have a specific date on which the order is required. They may want to say any time in June. There are infinite variations on this. So, do we want to reject entire orders because they cannot meet one specific criterion? It would seem to be better to treat all fields as text, process any orders that meet criteria, and put those that do not in a work queue.

Another concern is that there are limits to what can be validated on an XML document. While it is possible to set dates and such, it would not be reasonable to check for valid product codes, authorization to order, contractual ordering requirements, and so on. Therefore, some of the validation could end up in the XML and some in the back-end processing. A worst-case scenario here is that the validations are inconsistent.

In any complex system, structured data will exist on disk, on a screen or report, in memory as a business object, or in transmission as a business document. Business documents have been verbal, paper, EDI, and most recently, XML. When two cooperating organizations are communicating through business documents, they both need a flexible mechanism to communicate. It seems to me that putting validation in the Schema for an XML document does not allow for this flexibility.

mwacey@csc.com

Author Bio

Michael Wacey is a partner with CSC Consulting and has been involved in the data processing industry since 1982. He has worked as a CTO, CIO, and Project Leader in numerous areas, including the telecommunications, pharmaceutical, chemical, and financial industries.

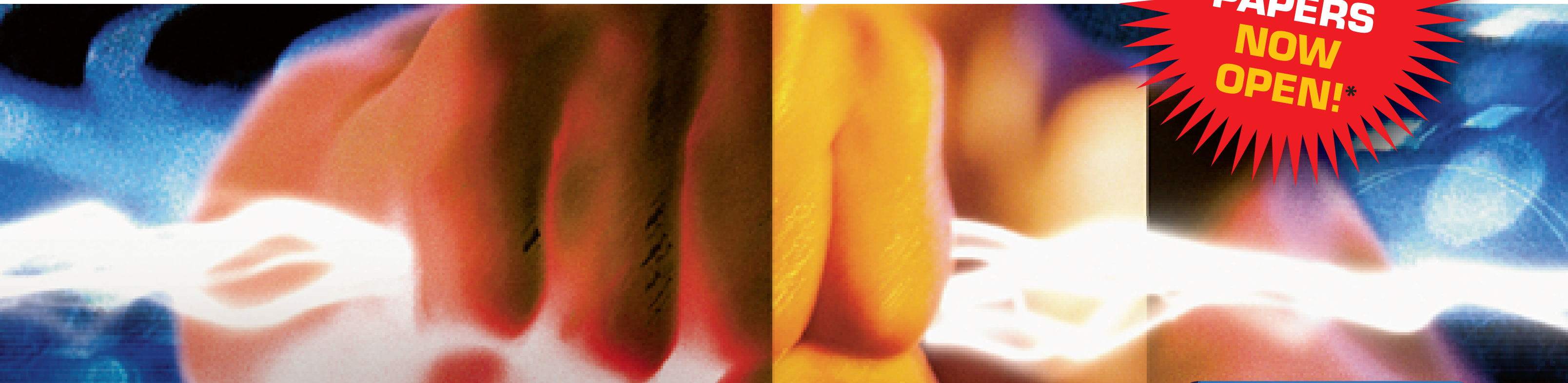
[Engage and Explore] the technologies, solutions and applications that are driving today's **Web services** initiatives and strategies...

www.sys-con.com/edge2005

web services EDGE conference&expo

FALL SERIES

CALL FOR PAPERS NOW OPEN!*



Coming to a City Near You ►

Web Services Edge

Fall Conference Series

3 Dynamic Conference Programs

Targeting Major Industry Markets

20+ seminars within 5 tracks will address the hottest topics & issues:

- ► Web Services: The Benefits and Challenges
- ► Web Services Security
- ► SOA (Service-Oriented Architecture) and ESB (Enterprise Service Bus) Strategies
- ► Interoperability, Incremental Integration, & Open Source
- ► The Management Process in Developing a Web Services Strategy

Program Features:

- ► Keynotes
- ► Tutorials
- ► Panel Discussions

Why Attend:

- ► Improve the return on your technology investment
- ► Develop & sharpen your strategy and identify key action steps
- ► Find new ways to reach and impress customers with Web services
- ► Maximize the power of your enterprise
- ► Protect your business from security threats
- ► Assess Web services as a viable option

Attention Exhibitors:

► An Exhibit-Forum will display leading Web services products, services, and solutions

Register Today! www.SYS-CON.com/Edge2005

Sponsored by

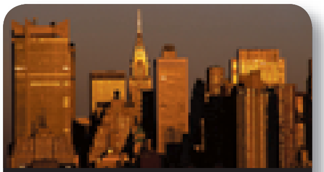
WebServices Journal · XML Journal · .NET Developer's Journal · eclipse developer's journal · WebSphere Journal · Information Storage+Security journal · wldj · JDJ · LinuxWorld Magazine · MX developer's journal · asp.netPRO · SD Times · CoDe · Software Test & Performance

*Call for Papers email: grisha@sys-con.com

For Exhibit and Sponsorship Information ► Call 201 802-3066

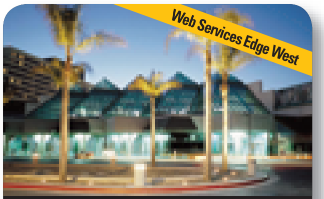
1

The Westin Washington, D.C.
Washington, D.C.
September 7-8, 2005



2

Helmsley Hotel
New York, NY
September 19-20, 2005



3

The Westin Santa Clara Convention Center
Santa Clara, CA
October 24-25, 2005

Produced by SYS-CON EVENTS

WRITTEN BY **Michael Sick**

Overview and Analysis of the Department of Defense Discovery Metadata Specification (DDMS)

How the DoD is using metadata to make its massive data stores more visible

The Department of Defense (DoD) Discovery Metadata Specification (DDMS) describes the DOD's preferred approach for decorating data assets with metadata. By providing a common convention for metadata, the DoD is building a common system for asset discovery, search, description, consumption, and security. This article provides a summary of the DDMS's purpose, structure, and capability. Upon completion the reader should have a basic understanding of the DDMS and should know where to go to get more detail and related materials. All questions regarding this article should be directed to Michael Sick at mike@serenesoftware.com.

AUTHOR BIO

Michael Sick is an independent J2EE and SOA architect who helps his clients solve complex product definition and design problems. He has more than 10 years of experience in the construction of distributed information systems and network technologies, having held positions from senior developer to senior systems architect to VP development. His work has crossed many domains including insurance, defense, finance, graphics & imaging, membership management, travel & entertainment, and e-commerce. He holds degrees in geology and political science from Guilford College, Greensboro, NC.

Metadata

Metadata is typically defined as "data about data." A metadata specification is an attempt to describe the format and content of a metadata convention. Establishing unambiguous conventions for the expression of metadata allows Communities of Interest (COIs) to better leverage their data assets, by making them more visible. Metadata conventions are beneficial because they can aid in the following areas shown in Table 1.

The DDMS establishes a broad set of categories for its metadata as well as a common set of data elements within the larger categories. While the DDMS provides several options for the metadata format (text, HTML, and XML), the formats are clear enough to support consistent metadata production, consumption, and validation. For an alternate look at metadata specifications, see BEA & IBM's ongoing efforts to establish a metadata specification for business computing at: http://dev2dev.bea.com/pub/a/2004/12/emd.html.

DDMS Overview

The United States Department of Defense governs the Army, Air Force, Navy, and a host of supporting intelligence and logistics agencies. Collectively the supporting IT organizations comprise one of the largest, if not the largest, IT enterprises in the world. The DoD's systems are highly variable in their implementation details and the DoD requires a comprehensive approach to making its data assets visible.

> "DDMS provides a comprehensive and extensible system for publishing metadata about data assets"

In May 2003, the DoD published the *DoD Net-Centric Data Strategy* that broadly defines the goals and approaches for making data assets available. The DDMS is the official response to the DoD's network-centric view of data discovery and descriptions and is designed to work across varying data formats, types, locations, and classifications. The DDMS specification is governed by the Global Information Grid (GIG) Enterprise Services Metadata Working Group (GES-MWG). The GES-MWG is charged with evolving the specification to continually match the concepts of operations adopted by DoD.

Consumption	Radically lower the costs of discovery and search mechanisms.
Production	Leverage common tools/COTS packages for the production of metadata.
Validation	Easily validate that required elements are present and conforming.

Table 1 •

Granularity

The DDMS specification is currently aimed at the higher-level data assets. The DDMS specification does not require that assets be described at the lower "record set" levels. However, the specification is flexible enough to allow tags to be placed at lower levels in the data hierarchy, if the implementers choose to do so.

Structure

DDMS content is separated into Core and Extensible layers. The Core Layer has four predefined element categories: Security, Resource, Summary Content and Format. Each element is assigned an obligation level (Mandatory, Mandatory Unless Not Applicable, Conditional, and Optional). The Extensible Layer is provided to contain content for domain-specific content areas. Additional obligation sets can be expressed through data requirement languages such as XML Schema and additional schemas can be registered in the DoD Metadata Registry.

The core layer is separated into four distinct category sets, Security, Resource, Summary, and Format, each containing elements supporting its designated role. The Security Set provides security-related information intended to classify the document. An external access control system can consume this information and authorize a client to view some or all of the DDMS description or underlying data. The Resource Set contains elements that provide ways to describe administration, maintenance, and pedigree of the data asset. The Summary Content Set describes elements most often associated with data searches and contains elements such as subject, title, and description. The Format Set describes physical aspects of the underlying data such as mime-type. Table 2 shows the Primary Category Elements for the Core Layer.

While the Security elements do not actively protect the data in the DDMS document, they do provide enough information for an external security system to make a decision on what information to provide based on the client's role and credentials. The samples provided in Listings 1-4 were derived from the specification and are intend to give the reader a feel for what a DDMS document might look like.

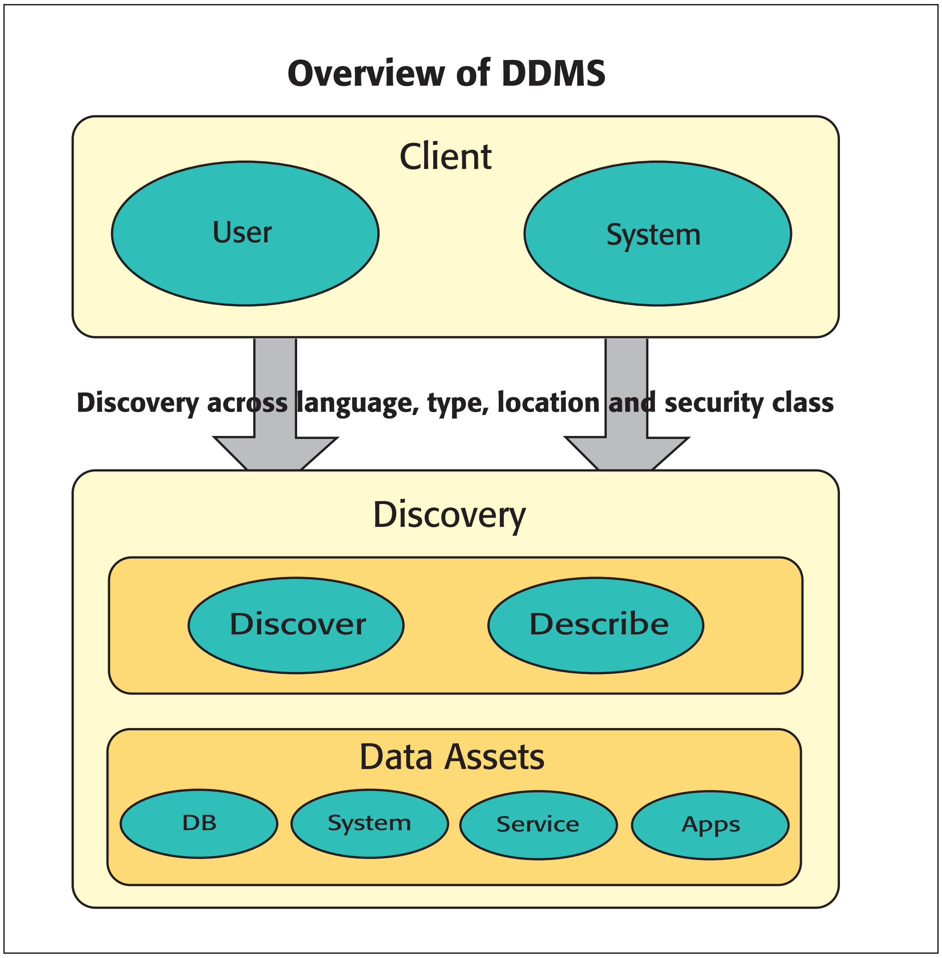


Figure 1 • Overview of DDMS

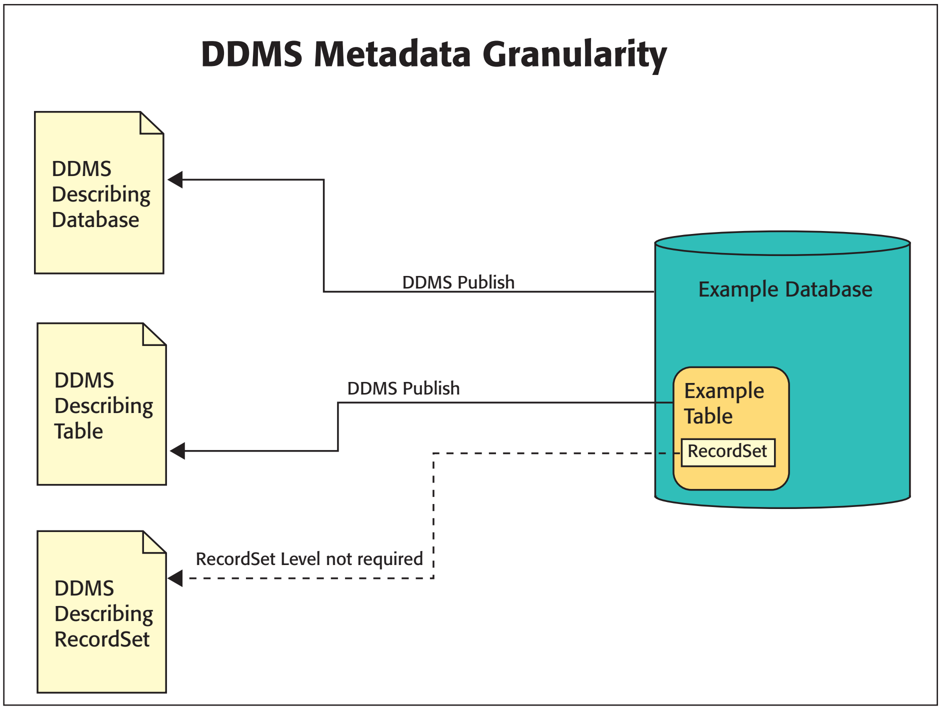
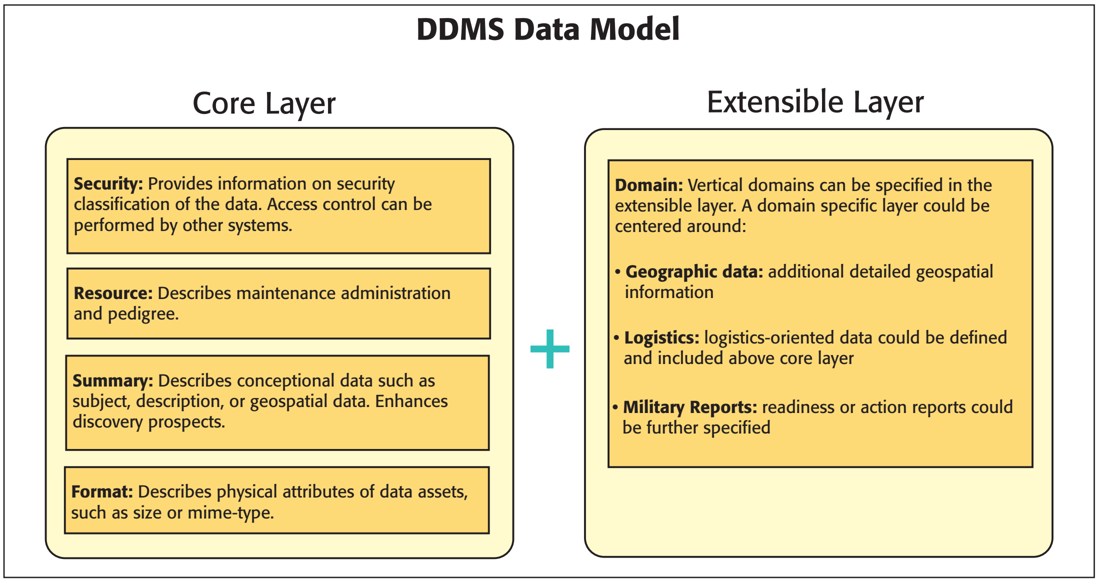


Figure 2 • DDMS Granularity



Security: Provides information on security classification of the data. Access control can be performed by other systems.

Resource: Describes maintenance administration and pedigree.

Summary: Describes conceptional data such as subject, description, or geospatial data. Enhances discovery prospects.

Format: Describes physical attributes of data assets, such as size or mime-type.

Domain: Vertical domains can be specified in the extensible layer. A domain specific layer could be centered around:

- **Geographic data:** additional detailed geospatial information
- **Logistics:** logistics-oriented data could be defined and included above core layer
- **Military Reports:** readiness or action reports could be further specified

Figure 3 • DDMS Data Model (modified C2.F2 diagram)

Core Layer Category Set	Primary Category	Obligation
Security	Security	Mandatory
Resource	Title	Mandatory
	Identifier	Mandatory
	Creator	Mandatory
	Publisher	Optional
	Contributor	Optional
	Date	Optional
	Rights	Optional
	Language	Optional
	Type	Optional
	Source	Optional
Summary	Subject	Mandatory
	Geospatial Coverage	Mandatory unless not Applicable
	Temporal Coverage	Mandatory unless not Applicable
	Virtual Coverage	Optional
Format	Format	Optional

Table 2 • Primary Category Sets (slightly modified C2.T1 diagram)

Suggestions

The core DDMS specification is clear and the site contains many resources (some protected) that help implementers quickly understand the technology. What was not evident from the site was if the DoD provides any resources and encouragement for Independent Software Vendors (ISVs) to implement DDMS support in their projects. Developers of relational database systems (RDBMS), content management systems (CMS), enterprise application integration tools (EAI), software modeling tools, enterprise resource planning (ERP) systems, and enterprise information integration (EII) tools should all be strongly encouraged to support DDMS. Custom programming of DDMS support is valuable but repeated implementations across DoD will certainly reduce ROI and slow implementation. A more visible program for ISV support would help the specification succeed over time.

Summary

DDMS provides a comprehensive and extensible system for publishing metadata about data assets. The organization of the standard into a minimal Core Layer and an Extensible Layer allows implementers to start with a subset of the information that they would ideally like to publish and then organically improve the quality of their metadata systems. The organization of elements into Security, Resource, Summary Content, and Format category sets allow producers and consumers to separate out design concerns for access, categorization, search, and consumption. As DDMS is incorporated into custom systems and commercial tools (RDBMS, modeling tools, EAI tools, EII tools, ERP systems, search engines) its value will continue to grow.

References

- *DDMS Site*, Global Information Grid (GIG) Enterprise Services metadata Working Group (GES-MWG): http://diides.ncr.disa.mil/mdreg/user/DDMS.cfm
- *Department of Defense Net-Centric Data Strategy*, Version 1.0, May 9, 2003, signed by the DoD CIO
- *Dublin Core Metadata Element Set*, Version 1.1, July 2, 1999, published by the Dublin Core Metadata Initiative (DCMI)
- *Enterprise Metadata Discovery*, published by BEA Systems: http://dev2dev.bea.com/pub/a/2004/12/emd.html

mike@serenesoft.com

Listing 1: Security Element

```
<ddms:security IC:classification=”U”
  IC:ownerProducer=”GB”
  IC:classifiedBy=”This Guy, Position Title”
  IC:SCIcontrols=”ST”
  IC:disseminationControls=”REL”
  IC:releasableTo=”USA GB” />
```

Listing 2: Example Resource Category Set Elements

```
<ddms:title
IC:classification=”U”
 IC:ownerProducer=”AUS”>
10 Wonders of Australia
</ddms:title>

<publisher
qualifier=”POC” IC:classification=”U”
IC:ownerProducer=”aus”>
<Person>
  <Surname>Smith</Surname>
  <Name>John.</Name>
  <UserID>abc123</UserID>
  <Affiliation>Boy Scouts</Affiliation>
  <Phonenumber>555-1212</Phonenumber>
  <Email>smith@smithouse.com</Email>
 </Person>
</publisher>
```

Listing 3: Example Summary Content Category Set Elements

```
<ddms:subjectCoverage>
 <ddms:Subject>
  <ddms:category
   ddms:qualifier=”http://thisplace.com/xml/”
   ddms:code=”A”
   ddms:label=”ASPEN”/>
 </ddms:Subject>
</ddms:subjectCoverage>

<GeoSpatial>
    <Region>Mid-Atlantic</Region>
    <PlaceName>The White House</PlaceName>
     <AddressLine>Middle and Main SStreets</
       AddressLine>
     <City>Washington</City>
     <State>VA</State>
     <CountryCode vocabulary=”ISO-3166”>USA</
       CountryCode>
     <PostalCode>22747</PostalCode>
</GeoSpatial>
```

Listing 4: Example Format Category Set Elements

```
<Format>
     <MediaFormat>text/XML</MediaFormat>
     <Extent qualifier =“sizeBytes“>5000</Extent
</Format>
```

XML JOURNAL
PRESENTS
XML
Resource CD
Secrets of the XML Masters
$198 VALUE
Inside:
SPECIAL
Every Issue of XML-J Ever Published on One CD!

SOA Software Introduces Industry's First Registry-Based Web Services Dashboard

(Santa Monica, CA) – SOA Software, a provider of comprehensive SOA and Web services management, security, and governance solutions, has announced the addition of the industry's first registry-based dashboard to its award-winning Service Manager. With registry-based dashboard's real-time alerts and visual indicators such as live charts and color-coded graphs, enterprise customers can monitor service-level agreements, and security incidents and vital performance metrics, thus dramatically improving the ability to discover, manage, and troubleshoot Web services.

The SOA Software Service Manager enables customers to reap the benefits of Web services by securing, monitoring, and managing XML and Web services and by providing consistent centralized management of service-oriented architectures (SOAs). The new registry-based dashboard builds on Service Manager's capabilities, delivering powerful, real-time insight into the performance, health, and availability of Web services through an easy to see and use graphical interface.

The SOA Software Registry is a fully compliant v3.0 UDDI server offering several advanced capabilities, including federated search and security and service-level policy metadata management capabilities. Service Manager's use of its own UDDI registry for service metadata management uniquely centralizes service policy information in a standards-based database. SOA Software's enhanced UDDI support allows customers to optionally leverage any number of additional third party registries (such as IBM WebSphere 6.0) at the same time, in a federated manner, providing maximum flexibility for heterogeneous environments.

QNX Introduces New Web Services Framework for Embedded Applications

(Paris) – QNX Software Systems has introduced a new framework that enables standards-based Web services for embedded systems.

Designed for inter-application communication without human intervention, Web services technology enables access to any type of service, over any type of transport, on any type of platform, while using any type of programming language. The new QNX Web services software development kit (SDK) offers embedded developers the ability to implement WSDL, SOAP, and XML-based Web service applications in native C/C++, hiding the complexity of XML and protocol transformations from the developer.

The QNX Web services SDK is conformant with the Web Services Interoperability Organization (WSI) Basic Profile 1.0. Applications developed with the QNX framework can therefore interoperate with other standards-based implementations, such as Microsoft .NET. Using QNX's flexible framework, users can also choose to use the default Hypertext Transfer Protocol (HTTP), or opt for another, such as SIP.

The QNX implementation of Web services is designed from the ground up to meet the constrained resource requirements of embedded systems. The small footprint of the framework is ideally suited for embedded applications such as remote monitoring, event logging, and in-field update of applications.

SeeBeyond and Raining Data Align to Generate Standards-Based RFID Solutions for Supply Chain Collaboration

(Chicago, IL) – SeeBeyond, provider of the world's first fully integrated composite application network suite (SeeBeyond ICAN) for advanced integration and composite application solutions, and Raining Data, an XML Data Management, Xquery, and Web services technology provider, announced an intent to enter into a strategic agreement whereby the two companies will team up to provide solutions to manage and orchestrate the flow of data generated by Radio Frequency Identification (RFID) systems. As a part of the intended agreement, the companies intend to enable the integration of information from existing applications to generate standards-based collaboration among partners across the supply chain, powering the EPCglobal Information Service, EPC-IS.

Together the companies will provide a standards-based technology platform that collects, manages, and integrates RFID and sensor data with existing systems to enable composite applications such as track and trace, product recall, advanced shipping and receiving reconciliation, and total inventory visibility.

As companies struggle to unlock the ROI around RFID initiatives, the realization of a best practices framework to enable real-time information sharing is becoming fundamental to any RFID project. This sharing of data not only encompasses the enterprise applications within the four walls of a corporation, but also the business processes that span the extended enterprise. The EPC-IS in the context of EPCglobal standards, is an important evolving component of any supply chain that needs to be able to share information with partners through an open standards framework.

Forum Systems Web Services Firewall and Gateway Security Solutions to Support Second Generation Specification for Federated Identity

(Salt Lake City, UT) – Forum Systems, the leader in Web services security for threat protection and trust management, announced that it has joined the Liberty Alliance Project as a sponsor and full voting member. Liberty Alliance is a global consortium for open federated identity standards and identity-based Web services representing more than 150 companies, nonprofit, and government organizations from around the globe, including American Express, Fidelity Investments, AOL, France Telecom, General Motors, HP, IBM, Intel, Novell, Oracle, RSA Security, Sun Microsystems, and VeriSign.

Forum also announced that it will support the Liberty Alliance second-generation framework for identity-based Web services in both the Forum Sentry Web Services Security Gateway and the Forum XWall Web Services Firewall. The ID-WSF 2.0 specification has been extended to include support for SAML 2.0, specifically defining how SAML 2.0 assertions can be used to communicate identity information for interoperable identity-based Web services.

Forum's Liberty integration will enhance the Single Sign On (SSO) features already in the Forum Sentry and Forum XWall products to support Federated Identity. This capability extends Forum's current SSO support to allow customers to achieve secure authentication, authorization, and single sign on, both inside and outside of an organization.