XML JOURNAL

THE ULTIMATE XML ENTERPRISE RESOURCE

February 2003 Volume: 4 Issue:2

$6.99US $7.99CAN

0 09281 01314 3

03>

SYS-CON MEDIA

XMLJOURNAL

SYS-CON MEDIA

Celebrate XML's 5th Birthday in Boston

WRITTEN BY **HITESH SETH**

On February 10, 1998, a revolutionary Internet technology child was born into this world and given an acronym that has since become synonymous with i-technology itself.

Even though it enters only its sixth year this month, XML has had a widespread effect on the nature of generations of technology to follow.

One factual point: XML was really conceptualized much earlier than 1998; for instance, the earliest XML Working Draft that I remember dates back to November 14, 1996.

Milestones in the History of XML
- **February 14, 1998:** XML 1.0
- **October 1, 1998:** DOM Level 1
- **January 14, 1999:** Namespaces in XML W3C
- **February 22, 1999:** RDF
- **November 16, 1999:** XSLT 1.0 and XPath 1.0
- **January 26, 2000:** XHTML 1.0
- **November 13, 2000:** DOM Level 2
- **May 2, 2001:** XML Schema
- **June 27, 2001:** XML Base & XLink 1.0
- **October 15, 2001:** XSL 1.0 (including XSL-FO)
- **February 12, 2002:** XML-Signature
- **December 10, 2002:** XML Encryption & Decryption

Five years later, we now see XML everywhere in the world of i-technology. Be it application-to-application integration, Web services, enterprise portals, or mobile computing, we've seen a rise in the number of technical jargons (you know, markup language for anything you can think of). Not all of them have completed the journey, but a significant number of MLs and vocabularies have found real-world applications and have gone on to become the de facto vocabulary for their purpose, industry, or use.

Certainly, XML has given the world of i-technology cause for celebration. We tend to celebrate a Silver Jubilee (25 years) or Golden Jubilee (50 years), but I think given the pace of IT, 5 years is a more appropriate anniversary to celebrate. After all, we don't have clear insight into what the nature of technology will be in 2025, let alone 2050.

So, how do you plan to celebrate the success of XML? I recommend making sure your enterprise applications are XMLized. If you've done that and would like to share your experiences with fellow technologists, or if you'd like to learn about how to really utilize XML and Web services in your applications, celebrate by attending the upcoming Web Services Edge 2003 East - International Web Services Conference & Expo, held at Boston's Hynes Convention Center, March 18–20.

A serious i-technology event, the conference will present a good mix of tracks: XML, Web services, Java, and .NET. Of course, there is XML itself in its pure form; Web services, probably the biggest application of XML; and Java/.NET, the two leading application development environments for building XML- and Web services-based applications and services.

The sessions in the XML Track range from an executive overview, "XML for Managers," to sessions on XML Schemas, OASIS, XSL-FO, XML security challenges, applications in life science, enterprise application integration and mobile computing, the Semantic Web, XQuery, XPath/XSLT 2.0, and XML tools. You can find out more information and register for the conference at www.sys-con.com/webservicesedge 2003east.

Also at this important juncture of the development of XML, I would like to learn how you have successfully (or unsuccessfully) used XML and related technologies. What do you think have been the key milestones for XML development so far? And what do you think are important milestones that XML has yet to achieve? What do you think about the explosive growth of the number of vocabularies based on XML – too few or too many? The key focus area for *XML-Journal* in recent times has been real-world applications, and your feedback would be very useful, as I intend to dedicate a section to your comments. Let's let the real world know what's being done and what can be done using this exciting technology.

AUTHOR BIO
Hitesh Seth, editor-in-chief of XML-Journal and XML Track chair for Web Services Edge 2003 East – International Web Services Conference & Expo, is chief technology officer of ikigo, Inc., a provider of XML and Web services monitoring and management software.

HITESH@SYS-CON.COM

Ados Co., Ltd.

http://a-dos.com

Behind the Firewall

WRITTEN BY **WIM GEURDEN**
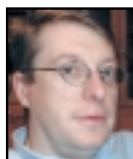
Questions about Web services and their uptake in financial services create a very black/white answer set. Some claim there's no usage; others say critical mass has been reached. The answer is somewhere in the middle.

One goal of the SOAP and UDDI message set was to let people discover services on the Web, and then connect to and start using them. We'd go shopping for mortgages and insurance policies by querying UDDI repositories and connecting to them. That hasn't happened. This vision was born during the dot-com craze and several aspects weren't defined: how to pay for such services, how to differentiate between them, and how consumers connect to them. The uptake of consumer-oriented Web services hasn't happened, which often leads to the conclusion that Web services have failed.

Web services also sought to offer better integration of the business processes of a value chain. Web services are being deployed to achieve easier integration, but this integration is mainly inside the firewall. Cross-company integration isn't happening on a large scale. There are two challenges with cross-company integration.

The first challenge isn't related to Web services but to resolving the business challenges associated with the tighter integration of any two companies. The second relates to the absence of the accepted standards that make any integration easier. To get to a broad standards acceptance, we must complete two sets of standards:

1. The core Web services infrastructure standards (related to messaging, security, and transactions), which are nearing completion.
2. Business-oriented standards in support of the processes to be integrated, which need definition. Industry bodies (IFX, OFX, ACORD, FSTC, etc.) are in the process of releasing Web services versions of their message standards.

This leaves us with Web services as a tool for internal integration. There isn't one financial institution that hasn't delivered a project using Web services. Here are some examples:

- Bank of Nova Scotia developed an ASP.NET application running on a Pocket PC connecting to a Web service that enables a VIN verification system to support their auto-dealer finance division. Due to real-time verification, the error rate will drop by 20%.
- Deutsche Bank's Global Equity Derivatives unit created a SOAP-based XML Web service for complex pricing logic running on their Solaris UNIX production environment to prove that XML Web services can be used to make Microsoft and other platforms interoperate. The development of new applications that require access to these pricing components has been dramatically accelerated due to the standardization surrounding these pricing engines.
- Travelers Property Casualty replaced their auto glass claim processing system with a Web services–based glass claims management system. This system reduces the overall claims processing time by 50% and lowers administrative cost by 20%.

Web services can lead to significant savings in development cost and significantly reduce the cost of the supported business processes by allowing for a greater degree of automation.

The future holds interesting opportunities for Web services. In retail banking, banking/call-center software vendors have created a Web services interface layer to business logic components, simplifying support for multiple channels. However, each vendor created their vocabulary. This helps with internal integration but doesn't allow for best-of-breed integration of various vendors. The ATM industry worked within the IFX forum to prevent such a fragmentation. Many of the upcoming IFX releases focus on ATM-related business processes. With IFX Web services mapping, we can envision the creation of Web services–enabled application servers accepting this IFX protocol. In wholesale banking we see a set of initiatives developing standards to integrate the accounts payable/receivable modules of ERP systems with the cash management systems of banks. The most exciting usage of Web services is in markets where the distribution of financial products is being split from the production of these products. In these markets rapid integration of new products in an existing delivery infrastructure, or the creation of a new distribution concept (e.g., franchised banking centers), will be key to survival. These business drivers will drive standardization between producers and distributors.

Every financial institution is experimenting with Web services, and most have small pilots in production. Until sufficient standardization happens in financial services–specific Web services standards, internal integration will be where the most success is achieved.

AUTHOR BIO

Wim Geurden is the senior technology strategist for banking at Microsoft Corporation. He works with banking customers, partners (software vendors and integrators), and the various Microsoft product groups to ensure that solutions are built that are scalable and reliable.

wimg@microsoft.com

HOME

Enterprise Solutions

Content Management

Data Management

XML Labs

Mindreef

www.mindreef.com

Reader Feedback

Is XML Complete?

I consider the current core specs of XML to be in an advanced state. However, it is clear that it's necessary to define more sophisticated techniques that aren't related to the SGML origins of XML. For example, in certain use cases it's useful to manage ID attribute types where the scope is similar to namespaces, e.g., it only operates on the content of an element or a document fragment; instead, the XML specs define that the scope of an ID attribute type is the document instance…this change might be considered in an XML 2.0 spec. I believe that XML is still not complete.

Sergio Rodriguez
srodriguez@canella.com.gt

What's Needed Is This

What I need (as a C programmer) is a way of getting all the data structures in my program written out to XML, then sucked back in again.

Here's the caveat: easily.

Surely someone's got some tools for taking C code–style structs and turning them into an XML file somewhere…but I can't find it (libexpat is halfway there, but still a pain – writing handler functions for a parser is great and all, but…).

All this talk of the wonder of XML and the benefits of using it as a store methodology is great and everything, but the "ease-of-use" factor isn't really there with XML as promised.

In the end, if you want to use XML programmatically, it's still just as much of a pain to use as if you were writing good-old C fread/fwrite-style save-/load-to-disk functions. The only difference is, with XML reading the results is a little easier than raw binary structs written to disk.

If I'm clueless about these things please feel free to beat me with a stick. It's just that libexpat and the like are a lot of overhead for someone who just wants to be able to save and load data structures into an XML file.

Jay Vaughan
via e-mail

Discovery of Complex Datatypes

WRITTEN BY **RON WEST**

Web services promises to change the landscape of distributed computing forever. Soon computer systems will freely exchange data independent of their language, architecture, or platform. Applications will be available on a per-use basis, and the current business model will be drastically modified as service-based computing expands the Internet into the next phase. In the technology adoption curve, we have just begun the "proof of concept" phase – a phase that will see the adoption rate of Web services increase steadily over the next 12 months. Shortly thereafter, the ability to link to systems written in different languages will be as common as HTTP itself.

The ability to connect disparate systems through a common data interchange is not the only quality that makes Web services so appealing. In fact, Web services discovery is really what sets the technology apart from existing systems. Although it is not completely developed, UDDI promises to deliver a common interface for discovery of available services. In order for dynamic discovery to be fully appreciated, Web services needs to achieve full maturation, which requires completely dynamic invocation. Having the ability to discover a service in an automated fashion offers little functionality if applications aren't flexible enough to routinely integrate the service into existing architectures.

When Web services first arrived on the scene, this type of flexibility seemed like a simple task. At the time we were dealing with "simple" services – services that returned basic data like stock quotes or weather data with strings and integers. Although these simple services could be dynamically invoked, they had no true business value. Now that simple services have evolved into more complex services with valid solutions for existing problems and contain more complex datatypes, like Amazon's integrated product search, the need for intelligent dynamic invocation has increased. This article will outline one particular solution for dynamic invocation of complex Web services. The article will also demonstrate the power of XPath, and its ability to move us toward a more interoperable Internet.

Note: This article assumes that you have prior knowledge of XML and XPath, and that you have read WSDL files and understand the architecture of describing the components of a Web service.

XML to the Rescue

Due to its native extensibility, XML was chosen as the platform for the Web Service Description Language (WSDL). Since WSDL is simply XML, the ability to qualify the Web services input and output parameters can be achieved using XPath. By using XPath to discover the service's properties, application frameworks that support XML and XPath can be enhanced through the systematic consumption of Web services. Consuming complex Web services can further enhance existing applications because it allows for multifaceted data exchange.

All Web services have a description file. The description file defines all of the interfaces to the service. It declares the service name, the

> ## "The nice thing about WSDL is that the XML Schema for the datatypes is often located within the XML file"

methods available, and the data format of all the input and output parameters. Since the process for discovering the major properties of the Web service will remain the same regardless of whether the service is complex or not, it is a good idea to determine if the service is complex. To do this, simply execute the following XPath search on the data from the WSDL file:

```
//*[contains(name(), 'complexType')]
```

Depending on the implementation of XPath, the above call will return some form of structured data, commonly arrays. If the result is positive, then the service has a method that either accepts complex data, returns data in a complex format, or both.

The following are XPath search strings that can be used to discover the main properties of a Web service. These steps remain the same regardless of the type of service being invoked.

HOME

Enterprise Solutions

Content Management

Data Management

XML Labs

1. *Service name:*
```
//*[contains(name(), 'service')]
```

2. *Available methods:*
```
//*[contains(name(),
'binding')]/*[contains(name(), '
  operation')][@name]
```

3. *Input/Output parameters for each
  method:*
```
//*[contains(name(),
'binding')]/*[contains(name(), '
  operation')][@name='#methodName#']/*
```

4. *Datatypes for each Input/Output
  parameter:*
```
//*[contains(name(),
'portType')]/*[contains(name(),
  'operation')][@name='#method
  Name#']/*[contains(name(),
  '#paramName#')]
```

The difference is what happens with the data from Step 4. In the case of a simple Web service, the datatypes are directly mapped to native datatypes and the properties of the service are stored accordingly. For complex datatypes, a bit more work is needed. The complex datatypes need to be further defined. The nice thing about WSDL is that the XML Schema for the datatypes is often located within the XML file. The properties just need to be extracted.

To extract the properties for the parameters, locate the schema. For instance, the Amazon Search Service has a method for retrieving data through a keyword search. The keyword search method has an input and an output parameter. The procedure to discover the properties of the input parameter is the same as the procedure for the output. To determine the properties of the input parameter of the keyword search, the following procedure can be performed:

1. First, discover the datatype of the parameter. Step 4 above will return "typens:KeywordRequest" as the value of the message attribute. The value of the message attribute defines the location for the datatype properties. The "typens" portion of the result refers to the namespace. In this case it is a local namespace, which means that the definition is defined locally within the file.

2. The "types" element contains all of the definitions for the extended datatypes. To discover the definition for this particular datatype, execute

the following XPath statement:

```
//*[contains(name(),
'types')]//*[@name='KeywordRequest']
```

The result returns the XML representation for the input parameter. Contained in this result are the elements required for a successful search.

3. Next, the element type is deciphered by retrieving the element name. In this case the element name is complexType, which means that this element may contain elements or may have additional attributes that further define its properties.

4. The next step is to traverse through the element to better understand the format of the data. The next element describes the overall content of the datatype. To retrieve the value of this element, execute the following XPath statement: /following-sibling::*. The name of the child element determines the process to be taken to ensure data integrity. Here is where things get a little tricky, as there are many predefined datatype formats that need to be understood. The formats contain definitions like "sequence" – a set of elements defined in a particular order – and "complexContent" – usually a reference to an array of elements. (To further understand all of the available datatype formats, look at the XML Schema definition at www.w3.org/XML/Schema). In the KeywordRequest datatype the format of this datatype is "all". This means that the children elements contained within this element can appear in any order, and must occur once and only once.

5. Because this is an element of type all, the next step is to collect the children elements that define the datatype. To do this, simply execute the following until all children have been retrieved: /descendant::*. Since each element can be used only once, only two pieces of data are important: the name and the type. Commonly, the type for these elements will be standard types like strings, dates, or integers.

6. The final step is to look for any attributes for the element. The attributes can be found by executing this code:

```
//*[contains(name(),
'types')]//*[@name='Keywor-
dRequest']/*[contains(name(),
'attribute'].
```

Attributes of the "attribute" element

define the conditional processing of the element's use. For instance, the element can be defined as required if the "attribute" element has an attribute named "use" with value "required". The attribute element is used for multiple purposes; this is just an example of one use.

The above process is just an example of how to discover the properties of a complex type. Each complex type has to be passed through an evaluator process with special case processing depending on its type. For instance, the data may be an array of elements. The array definition links to the element definition through an embedded complex type definition. A complete working knowledge of XML Schema is required to successfully consume a service with complex data. However, all of the rules for data definitions are easily accessible and often defined in-line.

"Complex-style Web services are easily defined using XML and XML Schema"

At this time, very few applications offer dynamic consumption of Web services. The applications that do support dynamic invocation can only invoke a limited set of Web services, due to the fact that dynamic invocation relies heavily on the loose coupling of data. That means that a process for describing Web services has to be abstract enough to offer multiple layers of data exchange, yet simple enough to easily integrate with existing software models. Complex-style Web services are easily defined using XML and XML Schema. However, in order for Web services to succeed, we need to begin to crystallize the original promises of dynamic discovery and invocation. Applications that currently support the invocation and publishing of Web services need to offer toolsets that seamlessly integrate services constructed with complex datatypes. ⊗

AUTHOR BIO

Ron West is a senior applications developer with PaperThin, Inc., a privately held Web content management vendor headquartered in Quincy, Massachusetts. Ron has been working with Web applications for seven years. He is one of the directors of the Rhode Island ColdFusion User Group, and is an established writer for several industry publications.

RWEST@PAPERTHIN.COM

Building a Real-World Web Service

– Part 1

WRITTEN BY
SUHAYL MASUD

RosettaNet makes the dream a reality

The answer is yes; Web services can be used to conduct real business, and this series of articles will show you how. In this installment, the first in a three-part series, I'll show you how to conduct an e-business dialogue, explain RosettaNet specifications, and create a Web service definition based on RosettaNet messages. Your organization can benefit from this real-world Web service, read on to discover how.

Are We There Yet?

Are widespread Web services a distant dream? While some analysts claim that lack of security and maturity are the reasons that real use of Web services is about 5–10 years away, I believe that real Web services are much closer, and they offer us great opportunity even today; we just have to be smart and creative about them. Rest assured, Web services are not just a vague notion, they can solve problems today, and they offer attractive features like the use of open standards such as WSDL and SOAP, and the reuse of existing Web infrastructure including hardware, software, and expertise. Most important, Web services are not venturing into unchartered territory; they benefit from lessons learned from tried-and-trusted e-business standards like RosettaNet and OAGIS.

However promising and appealing Web services appear, many organizations are hesitant to commit funds to Web service projects. "Show me the money," that famous line from *Jerry Maguire*, seems to be on every battle-weary manager's lips. This is a valid demand, and I believe we can provide this proof by building Web services that focus on reducing costs and saving time for existing business problems, rather than focusing on new business models enabled by Web services.

My goal is to build a real-world Web service that enables e-business dialogue between business partners. For this article, I'm applying Web services to solve the real business problem of purchase order management, a common activity in the business world and well suited for automation. The RosettaNet Web site (www.rosettanet.org) contains case studies that prove the benefits of implementing this activity as an e-business process. For example, Intel recently announced that it has used RosettaNet e-business processes to conduct over 10% of its selling and purchasing. That's more than $5 billion dollars worth of transactions. The Web service I'm constructing in this article is based on this proven RosettaNet specification, and you can easily show your manager "the money," the ROI, the business case, and anything else he or she might need.

Uncovering the True Value of Web Services

In the simplest sense, Web services allow us to define services with a very accessible and technology-neutral public interface. This interface is defined in XML-based WSDL (Web Service Description Language) and may be accessed (among other options) via the Web infrastructure using XML-based SOAP (Simple Object Access Protocol) messages. I will now explain why these simple features are causing so much excitement.

Web services are a revolution in the relationship between business and information technology. This revolution enforces a few points about e-business (see Figure 1):
• E-business is less about "e" and more about business.
• The public interface of an e-business puts services first.
• E-business dialogue is technology agnostic.
• Technology should assist e-business, not overpower it.

Web services are poised to bring significant change to how organizations think about e-business. While some of these changes are subtle, the results promise to be dramatic. The first change Web services bring to e-business is an intense focus on services, which is fitting, as one of the primary reasons an organization exists is to provide services to customers and partners. The second change is better communication and knowledge sharing between the different groups of an organization; Web services help everyone think "services," a shared concept. This enhanced communication helps in designing robust services and in getting the design right at the concept phase.

The most significant change in e-business is that Web services make business dialogue pretty much technology agnostic. With Web services, an organization provides access to its services through a public interface based on open stan-

Figure 1 • Web services bring profound change

dards and a common language. This separation between the public and private sides of the Web service removes technology concerns, such as OS, programming language, and platforms, from the e-business dialogue, enabling business partners with completely different technology setups for effortless communication. Removing technology considerations from the dialogue is the first step for conducting e-business dialogues; the next step is to use standardized dialogues, messages, and vocabulary (see Figure 2).

How E-Business Dialogues Work

The basis for a successful real-world Web service is the ability for business partners to conduct electronic business dialogue. Since we're talking about e-business, the dialogue is digital, and the software systems at both ends must make sense of it. A software system can really only "understand" what it has been programmed to understand, and the cost to conduct e-business dialogue can become prohibitively expensive if you have to program several interpretations of dialogues for the same business activity. For example, if you have 22 business partners with whom you buy and sell goods, if each of these partners defines a different dialogue for conducting purchase orders, you will have to implement 22 dialogues to do the same thing. This is why standard ways of describing, accessing, and processing business activities and e-business dialogue become crucial.

An e-business dialogue is the conversation that takes place between the software systems of business partners, to conduct a business activity. The dialogue can be simple or elaborate, based on the business activity that needs to be executed. For example, Acme and Laptops, Inc., conduct an e-business dialogue when Acme places a purchase order for 100 laptops with Laptops, Inc. An e-business dialogue can also be an elaborate composition of e-business dialogues. For example, Acme conducts an e-business dialogue to request marketing material from Laptops, Inc.; chooses some products and executes another e-business dialogue to request the price and availability of the chosen products from Laptops, Inc.; and finally executes an e-business dialogue to place a purchase order with Laptops, Inc. This entire sequence of business dialogues is itself part of a business dialogue.

In Figure 3, you can see a simple e-business dialogue in which Acme places a purchase order request with Laptops, Inc. You can also see the necessary components of an e-business dialogue:

- Open and accessible public interfaces
- Partner roles
- Standard messages exchanged in agreed-upon choreography
- Standard vocabulary
- An environment of security and trust

For Acme and Laptops, Inc., to start a business dialogue, both organizations need to have public interfaces that are defined in a standard manner and accessible via Internet-based protocols. The open and accessible interface makes it possible for Acme, which has a shop based on mainframes and COBOL, to communicate effortlessly and cheaply with Laptops, Inc., which is a Java shop.

To understand roles and choreography, think of a movie script. Just like a movie script, an e-business dialogue has different roles for the business partners to play. In our example, Acme assumes the role of a buyer and Laptops, Inc., assumes the role of the seller. Like actors delivering lines from a movie script, the organizations deliver messages to each other based
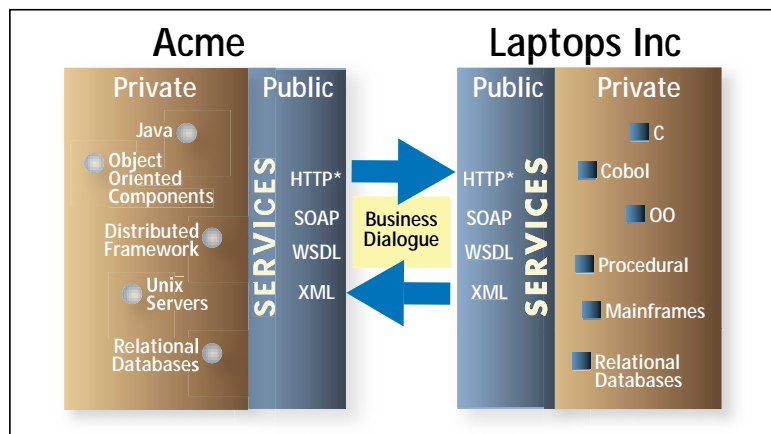

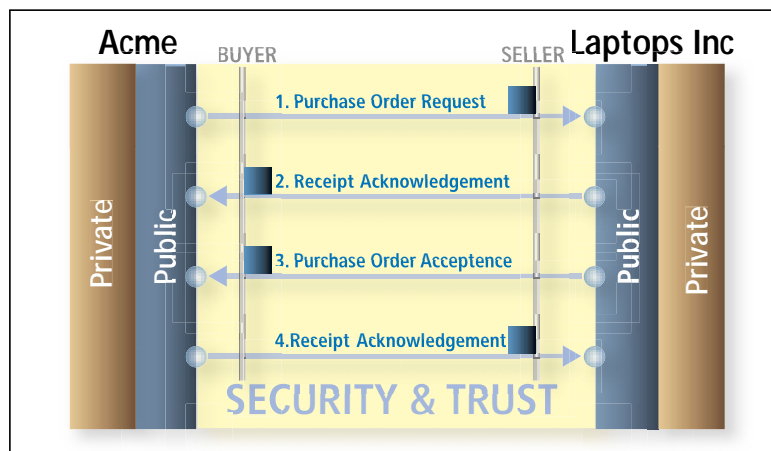**Figure 2** • The public and private sides of Web services

Figure 3 • The components of an e-business dialogue

on the order specified in the e-business dialogue. In e-business dialogues, this concept of sequence and order is known as choreography or orchestration. In the simplest form, it creates a sequence in which the messages are exchanged. The choreography can also be used to create elaborate dialogues that orchestrate services, partners, or other e-business dialogues. I will explain this in further detail in Part 2, which deals with BPEL4WS.

Keep in mind that the business dialogue is being conducted between software systems, and since they are not very good at guessing, nothing can be left ambiguous. The messages that Acme and Laptops, Inc., exchange have a standard structure and standard vocabulary. This means that the messages must follow a defined schema, and any words used in the message must have the same meaning to both organizations. Standardizing the definition and usage of a word is important, and we see this need even in our daily lives, as everyday words may take on different meanings in different professions. For example, when I mentioned a movie script earlier, if I had used only the word "script," it might have caused confusion about the meaning given our technical backgrounds – Perl script or movie script? Similarly, while "mouse" is a small, furry animal in a pharmaceutical lab, it is a pointing device in a computer lab. It is important that when a computer lab places a purchase order for a mouse, the seller doesn't ship a rodent. Luckily, there are standards bodies that build dictionaries to standardize definitions of words used in e-business messages.

One of the most important components of a business dialogue is trust and security. When organizations conduct business in the physical realm, they sign contracts to enter legally
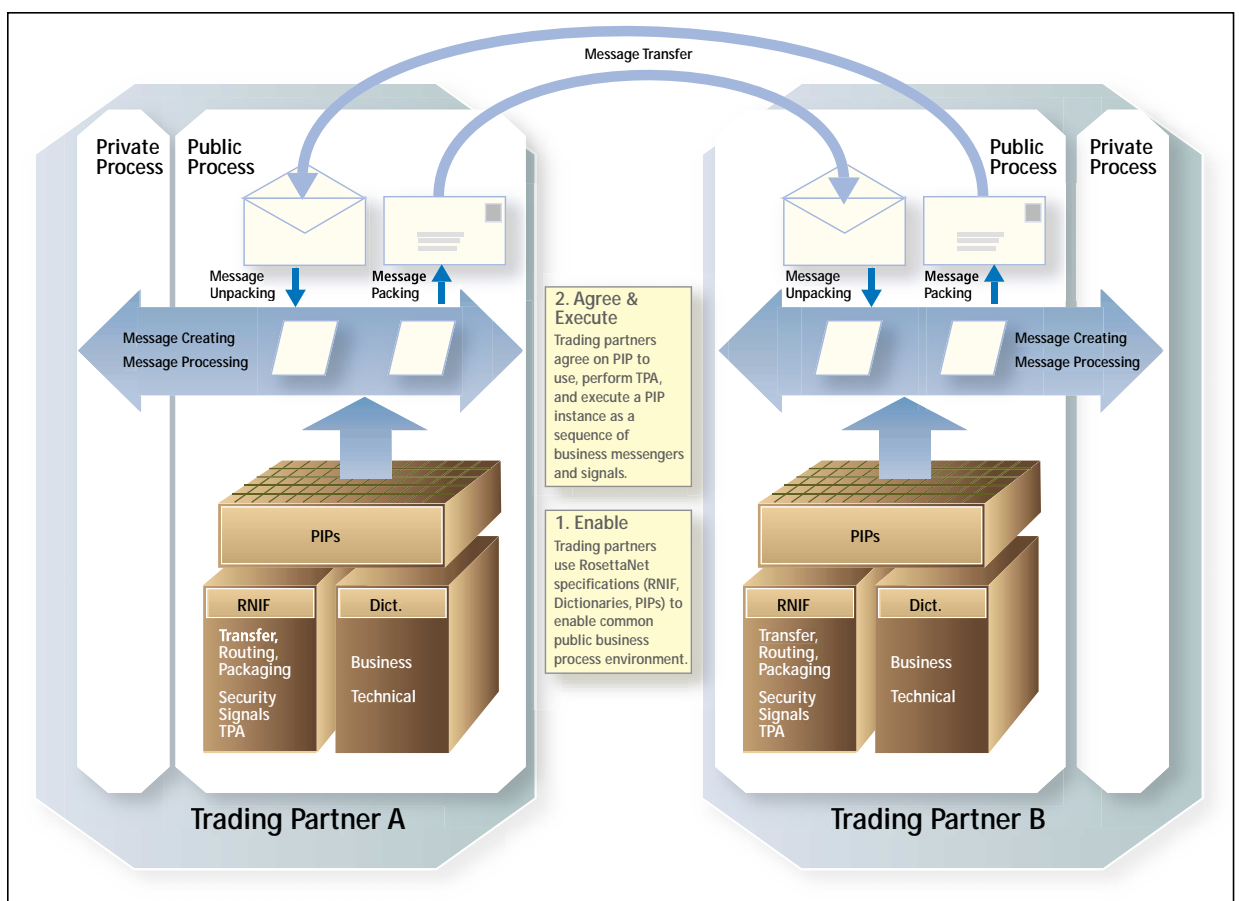
Figure 4 • How RosettaNet works (figure from RNIF 2.0)

binding agreements. In e-business, this trust is established by treating each exchanged message as a contract. The messages are stored for an agreed period, and each partner is bound to the promise made in the message. This concept is known as nonrepudiation. The partners also identify which employees are authorized to conduct the e-business dialogue, and authority and identity are verified during the e-business dialogue by inspecting digital certificates and the digital signature on the message. The messages exchanged are tamper-proof so that no third party can change the content and cause harm to either party. The messages are also transported over a secure wire to be snoop-proof, so no third party can eavesdrop on the sensitive information passing between the business partners.

What Is RosettaNet?

As I stated before, to create a real-world Web service, I need to enable an e-business dialogue. Along with open and accessible public interfaces, I need to use standardized messages, vocabulary, and choreography. Essentially, I need to take a standard e-business process and enable it in the Web services environment.

RosettaNet is an industry leader for e-business process specifications and therefore a natural choice for our real-world Web service. For the real-world Web service, I'll translate the RosettaNet PIP3A4, an e-business dialogue specification for placing purchase order requests, by modeling the messages in WSDL definitions and the choreography definition in BPEL4WS. One advantage of RosettaNet is that along with creating standards for vocabulary and messages, RosettaNet also standardizes the e-business dialogue. With previous standards, partners wishing to conduct a business dialogue would first have to see what messages they were capable of exchanging; then they would have to define their own dialogue by

determining what messages need to be exchanged, and in what sequence. With RosettaNet, this knowledge is standardized, enabling partners to conduct an e-business dialogue with very little setup time and lots of reuse (see Figure 4).

RosettaNet standards have three components:
1. PIPs (Partner Interface Processes)
2. Dictionaries
3. RNIF (RosettaNet Implementation Framework)

The PIPs are specifications for standard messages and choreography to conduct a specific business activity. The dictionaries standardize the words used in the messages and the RNIF contains specifications to enable trust and security in the message exchange. Our interest lies in selecting relevant components from RosettaNet and implementing them in the Web services environment. In this article I'll focus only on how to understand a PIP specification and which components we can use from it.

RosettaNet PIPs

RosettaNet PIPs create standard e-business dialogues for several business activities, such as order and inventory management, transportation, sales forecasting, and so on. The PIPs are organized in functionally logical groupings of segments and clusters. For example, the e-business dialogue we wish to conduct is PIP3A4 Purchase Order Request, found in the Quote and Entry Segment grouping, which belongs to the Order Management cluster. All PIPs are available for public download at the RosettaNet Web site.

The PIP specification package is a zipped file that contains three types of documents (see Figure 5). The specification itself is a Word document, help and guidelines for the messages are in HTML documents, and the message structure and content are captured in the XML DTDs. For the real-world Web
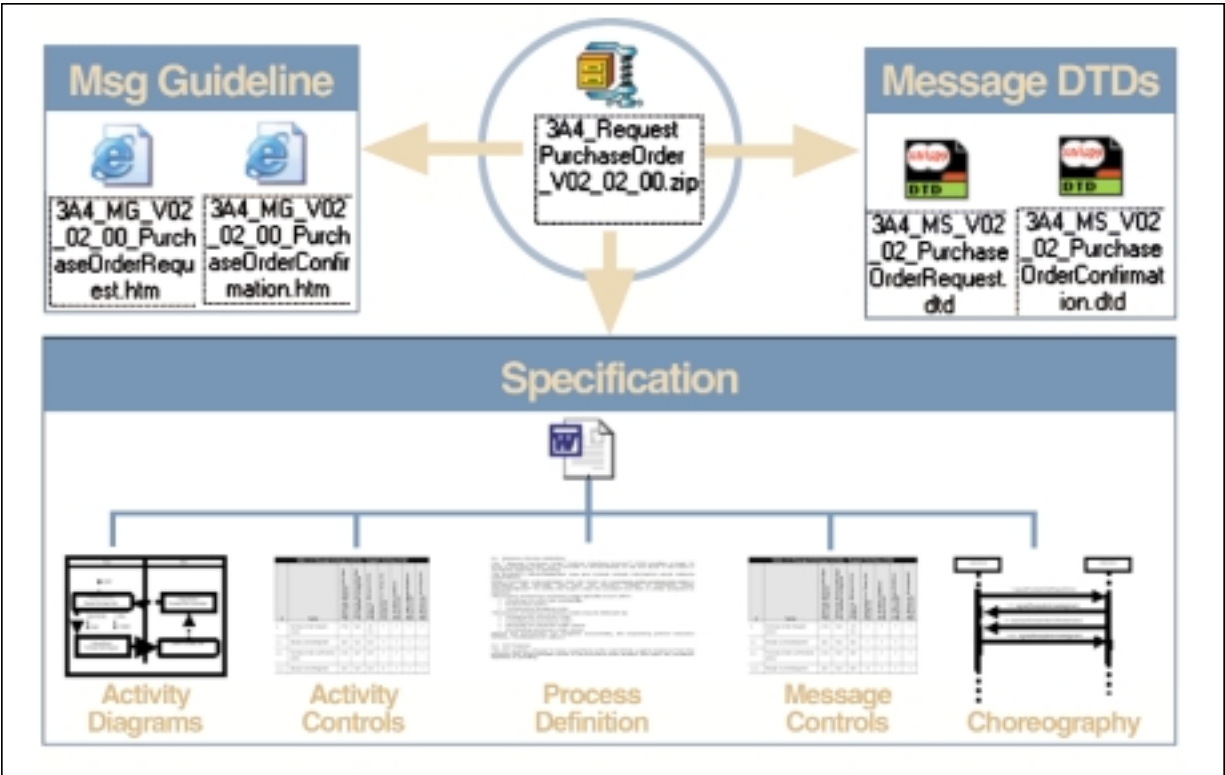
PolarLake

www.polarlake.com

Figure 5 • The PIP specification package

service, I'll extract the process definition and choreography information from the UML diagram and associated tables in the PIP3A4 specification Word document, and I'll use the DTDs to help create the messages and operations.

Figure 5 shows how the Request Purchase Order PIP works. The partners exchange two messages and two receipts; the two messages are the purchase order request and the purchase order acceptance, with receipt acknowledgment signals in between to signal the receipt of messages.

I want to model the messages as WSDL operations. What's missing from the PIP package is the DTD for the receipt messages. Since this message is common to all PIPs, it's part of the implementation framework and I will pull it out of the RNIF package (also available on the RosettaNet Web site).

To model the PIP receipt and messages as WSDL operations, I need to convert the DTDs to XML Schema documents;

this is a simple task since most XML authoring tools can perform this conversion for you. However, there is a small problem with the DTDs in the RosettaNet specification, due to an improper object model of the elements, and each DTD appears to be stand-alone and not part of a bigger model. Therefore, elements are at times duplicated or redefined in different DTDs. While this is not a big problem in the RosettaNet environment, in the Web services definition it becomes a problem, as the schemas from these DTDs need to be combined in the same document. RosettaNet has promised that these problems will go away when they roll out a remodeled, well-connected schema. In the two DTDs I converted to schema, I manually renamed a couple of objects and changed references to fix this problem. Having extracted the messages I want to use in our e-business dialogue, I need to translate these messages into Web service operations. The next step will be to define the business dialogue as a BPEL4WS definition.

Translating RosettaNet into WSDL

It is now time to work with WSDL, the language used for defining and describing Web services. WSDL is based on XML, and has a simple, easy-to-learn grammar. A WSDL file defines the types, messages, and operations of a Web service, as well as the message format and protocol information for these operations. The main components of a Web service definition are:
• Types
• Message
• Operation
• Port Type
• Binding
• Port
• Service

I'll explain each component further as I flesh out the WSDL definition for the real-world Web service.

Figure 6 • The choreography diagram from PIP3A4

I first want to construct the type element of the WSDL definition. The type element is where all the data type definitions are described; these data types are later used to describe the messages in the message element. I have extracted the message DTDs from the RosettaNet PIP3A4, and after converting them to schemas and fixing the duplication problem, they are placed in the type element. Listing 1 shows what a type definition looks like.

Next, I'll create the message elements of the WSDL definition. Messages defined in this section are used within operations. Each message is composed of one or more parts, and each part is associated with a type defined earlier. I will define the following messages:
- Purchase Order Request
- Purchase Order Response
- Receipt Acknowledgement
- Exception Message

Here are a couple of the WSDL message definitions:

```
<wsdl:message name="placePurchaseOrderRequest">
 <wsdl:part type="typens:Pip3A4PurchaseOrderRequest"
name="purchaseOrderRequest"/>
</wsdl:message>
<wsdl:message name="sendPurchaseOrderConfirmation">
 <wsdl:part type="typens:Pip3A4PurchaseOrderConfirmation"
name="purchaseOrderConfirmation"/>
</wsdl:message>
```

Now that I've created the messages that will be used in operation elements, it's time to think of how to define the operations and port types. A port type is a set of operations, and an operation is a pattern for exchanging messages. In WSDL, you can define four types of operations:
- **One Way:** The service receives a message
- **Notification:** The service sends a message
- **Request/Response:** The service receives a message and it replies with a message
- **Solicit/Response:** The service sends a message and receives a response

Looking back at the choreography of the PIP3A4 in Figure 5, Laptops, Inc., receives a request from Acme and sends back a receipt acknowledgement. Then Laptops, Inc., sends an acceptance to Acme and receives a receipt. Since I'm implementing this service assuming the role of the seller, and the WSDL specifications indicate that WSDL currently only contains bindings for the One Way and the Request/Response, my choices for implementing the operation are limited to a Request/Response operation. This means that I'll have to forget about the receipt acknowledgement signals for the time being. With this limitation, here is how to define the port types:

```
<wsdl:portType name="QuoteAndOrderEntryPort">
 <wsdl:operation name="requestPurchaseOrder">
  <wsdl:input message="typens:placePurchaseOrderRequest"/>
  <wsdl:output message="typens:sendPurchaseOrderConfirma-
tion"/>
 </wsdl:operation>
</wsdl:portType>
```

The next step is to define the bindings; this part describes the message format of the service as well as the protocol details for the different messages and operations. WSDL includes bindings for SOAP 1.1 and HTTP GET and POST. I will use the SOAP binding for the real-world service. Listing 2 shows how it is described.

To wrap things up, I'll define the service element here:

```
<wsdl:service name="QuoteAndOrderEntryService">
 <wsdl:port binding="typens:QuoteAndOrderEntryBinding"
   name="QuoteAndOrderEntryPort">
  <soap:address
location="http://www.DifferentThinking.com/xmlj_article/
  soap"/>
 </wsdl:port>
</wsdl:service>
```

We now have the WSDL definition of our real-world Web service. In Part 2, I'll show you how to use BPEL4WS to compose a full-fledged business dialogue using the WSDL definition from this article. ⊗

AUTHOR BIO

Suhayl Masud is a lead consultant at Different Thinking; he has over eight years of experience in the software industry working with object-oriented technologies, Java, and XML and constructing architectures for e-business systems. Suhayl has consulted as the lead technical architect at Roset-taNet, where he helped develop the next generation of e-business specifications.

SUHAYL@DIFFERENTTHINKING.COM

LISTING 1 •

```
<wsdl:types>
     <xsd:schema
targetNamespace="urn:QuoteAnOrderEntry"

xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:xsd="http://www.w3.org/2001/">
  <xsd:element name="DateStamp"
type="xsd:string"/>
  <xsd:element name="DateTimeStamp"
type="xsd:string"/> …
 </xsd:schema>
</wsdl:types>
```

LISTING 2 •

```
<wsdl:binding name="QuoteAndOrderEntryBinding"
type="typens:QuoteAndOrderEntryPort">
 <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
 <wsdl:operation name="requestPurchaseOrder">
  <soap:operation soapAction="urn:QuoteAndOrderEn-
tryAction"/>
   <wsdl:input>
<soap:body
encodingStyle=http://schemas.xmlsoap.org/soap/enco
ding/ namespace="urn:QuoteAndOrderEntry"
use="encoded"/>
   </wsdl:input>
   <wsdl:output>
<soap:body
encodingStyle=http://schemas.xmlsoap.org/soap/enco
ding/ namespace="urn:QuoteAndOrderEntry"
use="encoded"/>
   </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

▼ Download the Code
▼ www.sys-con.com/xml

WRITTEN BY **ENRIQUE CASTRO**

Business Systems with Web Services

Not just bells and whistles

Today Web services is still an evolving and emerging technology. This article covers a number of use models, from the perspective of both the developers building systems that incorporate Web services, and the user's experience with these systems.

Web services represents an evolutionary development of the original, browser-based Web developed at CERN in the early '90s. The original Web assumes the existence of a human interacting with a browser and a Web server that responds to browser requests over a network. Web services adds a number of standards and protocols that allow computers in a network to interact autonomously, driven by programs, in lieu of browser and server interactions. The loosely coupled, architecture-independent characteristics of the original Web are preserved. Some essential standards include:

- *Extensible Markup Language (XML):* A universal data format
- *Simple Object Access Protocol (SOAP):* A protocol to convey XML
- *Universal Discovery, Description, and Integration (UDDI):* A method for the different Web services to find each other
- *Web Services Description Language (WSDL):* An XML-based language used to describe the goodies available from a specific Web service

The purpose of these standards and protocols becomes obvious once you realize that the Web services–specific standards mimic their browser-based counterparts – XML is used as the data format for machine-to-machine interactions instead of the HTML format used in the traditional Web. HTML contains annotations to instruct the browser on how to display the data within. These annotations would be superfluous in machine-to-machine interactions. SOAP is a protocol that allows XML data to be encapsulated inside HTTP messages. Strictly speaking, there is no requirement that HTTP be used. It's perfectly possible to do the encapsulation inside mail (SMTP, or Simple Mail Transfer Protocol) messages. This possibility has been mentioned in the literature, although it is not in use today. Finally, UDDI and WSDL mimic the functionality of a Web search engine. It is significantly easier to write a program that uses UDDI and WSDL than a program that parses the unstructured text returned by Web search engines.

Practical Benefits of Web Services

The loosely coupled nature of Web services allows the components to be separately developed and can radically reduce the time it takes developers to have an application up and running. A change in a distributed API in a traditional, tightly coupled application requires that every running copy be replaced to reflect the change, or extensive coding and testing to ensure backward and forward compatibility. Alternatively, Web services allow the binding between a service provider and a requester to be done at the time of the request, not at design time. This mode of interaction is called *late binding*. Late binding is beneficial not just to developers of new applications; it's now feasible to retrofit older applications to work in a Web services environment through the use of thin software proxies without any changes to the original application, a very useful feature when a design goal is to not disturb existing functionality.

The power of Web services cannot be measured in terms of the "bells and whistles" it brings, but in the mundane ways it allows people to accomplish their work, quietly and unobtrusively. Developers now can integrate functionality in a fraction of the time it would have taken within a more traditional homogeneous object framework using compiled languages, and the user community benefits from the new capabilities. The use of industry standards allows highly heterogeneous components to work together, reducing implementation cost and obviating the need to use expensive proprietary middleware.

From an end-user perspective, Web services can be transparent, with users unaware that the technology has been incorporated in a specific application. Web services has the potential of reducing retraining by allowing the repurposing of existing systems. For instance, staff in a small but growing company, accustomed to tracking their inventory with spreadsheets, may need to eventually track it with a database program. Ordinarily it would have been necessary to expose interactions with the database, or at least get the numbers through a browser and transcribe the data from and to the spreadsheets. If the database has a Web services interface, however, it might be possible to write a Web services–based connecting program that updates the same spreadsheets automatically, using existing processes and reducing the need for retraining.

Web Services and Client Applications

Let's take a look at the traditional three-tier infrastructure (see Figure 1).

AUTHOR BIO

Enrique Castro is a strategic technology architect and consultant who has been with Intel Corporation for 19 years. Enrique is a cofounder of Neighborhood Learning Center, a nonprofit organization providing education and tutoring services to K–12 students and senior citizens. He has MS degrees in electrical engineering and computer science and a PhD in electrical engineeering, all from Purdue University.

Enterprise Solutions · Content Management · Data Management · XML Labs · HOME

Little more than the Web server and browsers would be needed for the distribution of static data. This is rarely the case today for most nontrivial Web-enabled applications. The Web server officiates as a highly sophisticated application front end. It receives requests from a browser, finds the application it is for (there might be several), assembles the data from the browser request into the appropriate method invocation using a selected script or stored program, and sends the call through the internal network to the server running the application logic. The application is usually architected with the business logic and data running in separate computers. The data is placed in a database engine running in the back end. The business logic may be multilayered. When multiple database operations need to be linked together as an all-or-none operation, this function might be fulfilled by a transaction processing monitor that is part of an application server. The incoming request may trigger a significant amount of processing in the logic engine, and a number of changes ensue in the data stored in the database. The results of the operation are sent by the logic engine back to the Web server running in the front end. The Web server interprets the data returned from the logic engine and generates HTML on the fly, which is sent back to the browser.

The changes needed are subtle. HTTP requests are replaced by SOAP requests. There are no radical protocol changes since SOAP is essentially a method for conveying XML data embedded in the HTTP protocol HTML messages. The Web server needs additional capabilities, including the understanding of the new protocols particular to Web services. Not all the capabilities need to be implemented by the Web server. Take, for instance, the UDDI service. It might not be needed if the service components "know" each other already, and if they don't, the service could be provided by some other provider (see Figure 2).

The dynamics of this new setup are quite different. In a traditional Web setting the browser's main role is to convey the data from the Web server. There is little local computation on the client other than the processing of the formatting directives. Even when there is local computation such as JavaScript, or applications such as Macromedia Flash, the task is subordinated to dis-

playing data emanating from the server. The snippets of code that run at the browser are usually loaded from the server. There is very little independence between the client and the server. The client "application" is always a browser.

There is nothing inherently "wrong" with a browser interface. However, Web services presents developers with new and powerful architectural choices combined with reduced implementation cost as compared to the traditional Web, without sacrificing any of the benefits. The browser model, while offering a universal interface, requires performance compromises. Due to the stateless nature of HTTP, refreshing one field in a display requires that the complete page be refreshed. This means a round-trip back to the server with a heavy protocol can take anywhere from a few seconds to several minutes, depending on the connection speed, network, and server loading. The server may recompute other elements displayed in the page, complete with a trip to the database. Doing so is not efficient, but it is easier to implement than writing code to check consistency. This behavior is acceptable for occasional queries, for

whereby most computation is performed on the server side. For instance, Web service–based clients could facilitate the implementation of shopping carts. Because HTTP is stateless, traditional implementations require piggybacking the contents of the cart between the client and the server until the transaction is complete. The client application can be designed to hold the current state of the shopping cart, with the associated reduced network traffic and improved system response.

The reader is cautioned that the three-tier framework is a convenient architectural abstraction to make a very complex system understandable. Reality is much messier. The environment in which Web services are deployed is not usually a greenfield, where the architect or developer gets to design the system from the ground up. This is the environment where Web services technology offers the greatest potential because of the advantages of late binding.

Web Services and Business Logic

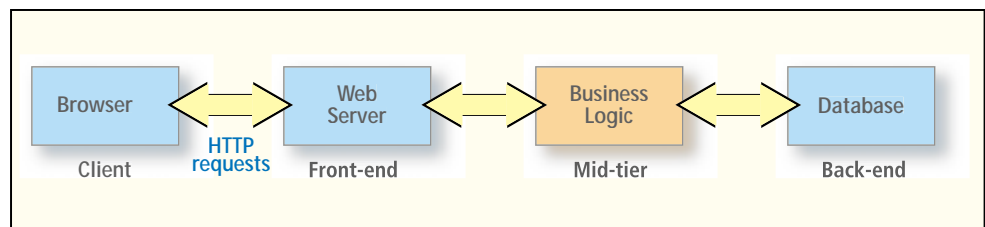The use of Web services to implement business logic is better explained through an example, in this case an

Figure 1 • Three-tier distributed infrastructure for Web-based applications
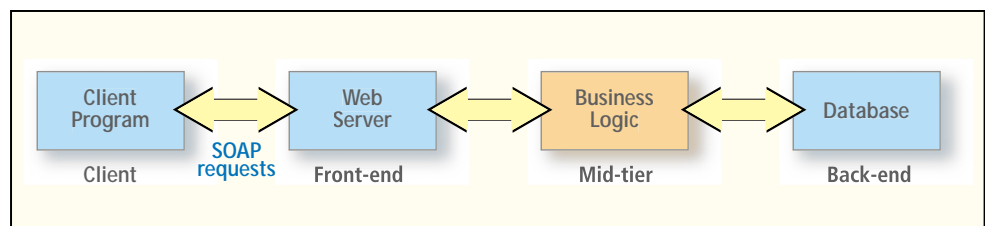

Figure 2 • Three-tier distributed infrastructure for Web services

instance, a user checking a mutual fund account balance. However, it is painful to use as an everyday application. This is why Web-based mail clients are used only under duress, when no other method is available.

Web services allows developers and system architects enormous freedom to design a distributed system. The minimalist approach is the traditional Web application environment,

implementation in the insurance industry. The system is used to generate vehicle insurance quotes and provides interfaces to independent agents, insurance brokers, and end users.

The communication styles used to implement business logic are much richer than in the client scenario studied in the previous section. The most prevalent scenario for client communications is the synchronous request/reply,
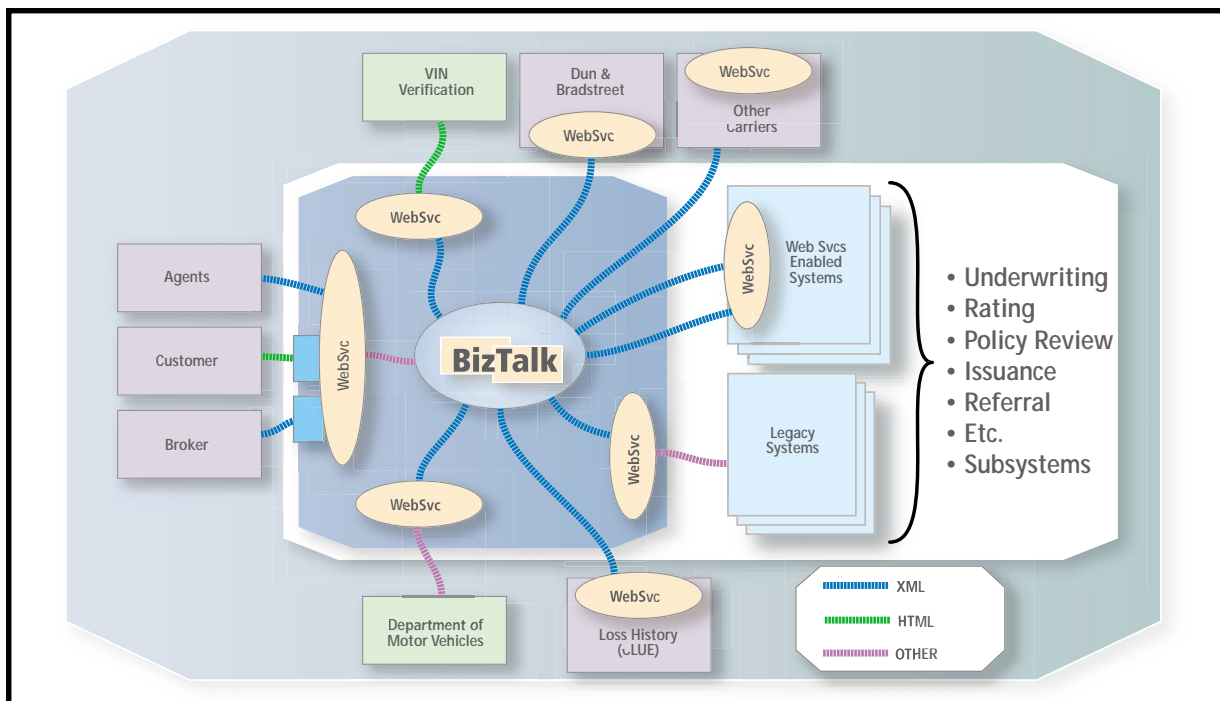
Figure 3 • Web services used to implement business logic

whereby the client sends a message and then waits for a response from the server. This is called "pull" technology because the interested party initiates a data exchange. Depending on the building blocks to be used, other communication modes can be implemented with Web services. These include publish/subscribe (push technologies) and various forms of asynchronous messaging. Asynchronous means that a requester does not wait for a reply before sending the next message.

Example: An Insurance Quote System Using Web Services

This example is drawn from an actual engagement of Intel Solution Services, a consulting organization of Intel Corporation, with an insurance company. Even though the components in the illustration are nominally part of the internal business logic, the functionality is built with components provided by external organizations. When a client requests an insurance quote from an agent for a particular vehicle, the agent submits the request. The business logic (partially implemented in Microsoft BizTalk) triggers secondary requests to Dun & Bradstreet to assess the client's risk. Other requests go to the Department of Motor Vehicles (DMV) to check the client's vehicle identification number (VIN) and driver's license. Prior loss records are checked against the CLUE (Comprehensive Loss Underwriting Exchange) database.

Business logic is implemented as a mixture of traditional building blocks and Web services–based building blocks nicely coexisting with each other. The business logic subsystem is continuously evolving, including the mix between traditional and Web services–based components. Traditional processes also include manual transcription of data. The system might have been enhanced at some point by providing the clerk with Web access to the same information. Web services can be used to streamline interfaces between external information providers and the internal business logic, reducing labor costs. The two systems may coexist for a long time.

The three-tier structure described here exists in this example, although not in an obvious form. The tiers do not exist as physically separate entities. Instead, the system is built as a swarm of collaborating subsystems. Each component subsystem may be a distributed system by itself, such as the "Web Services–Enabled Systems" block, each with its three-tier structure, ad infinitum (see Figure 3).

Conclusion

Web services augments the traditional browser-based Web with machine-to-machine interactions. Web services–based systems can provide traditional browser-based services as well as callable interfaces to be used by clients or to implement other services. Accessing a Web services–based system through a browser should be indistinguishable from access-

ing a traditional Web application with the same browser.

For Further Reading

For further insight into case studies involving Web services, there is an excellent article in *CIO Magazine*, (www.cio.com/archive/040102/real.html) "Web Services in the Real World."

IT industry trade magazines are recommended reading for the latest milestones in Web services technology.

Vendors have different approaches to Web services. A suggested starting list includes IBM Corporation; Microsoft Corporation; Sun Microsystems, Inc.; IONA Technologies; Oracle Corporation; and BEA Systems, Inc.

Standards are fundamental to the workings of Web services. Here is a seed list:
- *The World Wide Web Consortium:* www.w3.org
- *Organization for the Advancement of Structured Information Standards:* www.oasis-open.org
- *Web Services Interoperability:* www.ws-i.org
- *WebServices.Org:* www.webservices.org

For additional articles by Intel Solutions Services engineers, please visit www.intel.com/intelservices/intelsolutions.

ENRIQUE.G.CASTRO-LEON@INTEL.COM

BEA eWorld

www.bea-eworld.com

The Trouble with Tables

They're trickier than they seem

WRITTEN BY

NEIL BRADLEY

When the content of an XML document is intended for human eyes, rather than software digestion, it's necessary to consider the possible limitations of rendering applications, including typesetting systems, DTP packages, and on-screen document presentation tools such as Web browsers. Simple text structures, like headings, paragraphs, and list items can be presented easily enough, but intricate structures, such as chemical and multiline mathematical formulas, are so difficult to render that they're often preprocessed into images. Between these extremes we have tabular structures.

The trouble with tables is that they are too common and straightforward to be treated as images, yet they are often sufficiently complex to cause problems for rendering applications. Standard table models can help, and two such models are discussed in this article, though their more advanced features must be used cautiously. Semantic markup with dynamic conversion to tabular structures is another possible approach.

Complex Tables

A table is complex, and potentially causes problems for rendering applications, if it:
- Lacks column width information
- Includes columns of text that are vertically aligned on a specific character
- Includes cells that occupy more than one grid location
- Has complex cell content, possibly including embedded tables
- Includes repeatable header and footnote rows
- Includes border lines

Figure 1 shows a complex table as it might be presented to the document author in a sophisticated XML authoring package (note that such visual feedback helps authors avoid making mistakes). Figure 2 shows how this table might be formatted when the document it is contained within is presented on paper. All of the complexity factors listed above are demonstrated (with the exception of complex cell content), in part because the table happens to be split across two pages.

Complex Table Tagging

A table is theoretically simple enough, being merely a series of rows and columns that intersect to create cells, and XML elements can be employed to create this structure. Any number of strategies are plausible and, at first sight, the most significant decision when creating an XML table model (as part of a DTD or schema) is whether tables should be built row-by-row or column-by-column (in fact, the former approach is usually taken, and is the basis of the two standards discussed later):

```
<table>
  <row>
    <cell>...</cell><cell>...</cell>
  </row>
  <row>
    <cell>...</cell><cell>...</cell>
  </row>
</table>
```

XML tagging can also express the complexities listed above without difficulty. Typically, attributes are used to specify column widths; the presence and style of border lines; alignment of text within a cell; and the size of a cell that occupies adja-

cent cell spaces (a "straddling" cell). Element hierarchies can describe tables and other structures within a cell, and elements are also typically used to identify header and footnote rows. But simply adding elements and attributes to the table model in order to permit such complexities to be introduced does nothing to help with the rendering task (quite the reverse). Each of the problem areas listed provides a unique set of difficulties for any rendering application.

Column Widths

The first problem a document instance author might encounter when creating a table is the need to decide how wide to make each column, and there are several strategies to consider.

From the author's point of view, the simplest thing to do is to ignore the issue, and not bother to specify any column widths. The rendering application then has to decide how wide to make the columns, and might vary the widths depending upon how much text it finds in each column. But this is hardly an exact science, and the results are not always pleasing to the reader (or to the author). In Figure 2, the first column is narrower than it should be, considering the generous space given to the other columns. When there is a lot of text in each column, and limited horizontal space, the results may well be illegible. Another consequence of this approach is that table rendering in a browser is slow, because the browser must load the entire table, then analyze the content, before it can even begin to show the first row to the user. Note that specifying a width for the entire table, as some table models allow, would not help at all.

If a table is going to be presented once, on a single medium, with a known display or page width, then the author could reasonably be required to specify the exact width of each column in points, millimeters, or some other convenient measure. Pixel measures, however, are not recommended, because pixels vary in size depending upon the resolution of the screen or printer.

More flexibility is needed if multiple products are to be extracted from the same source data, or if the document is to be published on a variety of media. The size of the text in the cells and the available column widths may vary enormously. The author could focus on relative column widths instead, and specify proportional or percentage values instead of fixed values. For example, a column may need to be 10% of the total table width. However, some rendering tools require fixed units, in which case it becomes necessary to convert proportional or percentage measures into such values (for example, "10%" might be translated into "15mm" in one scenario, "22mm" in another, and "30pt" in a third).

Complex Text Alignments

Apart from the usual left, right, center, and justified alignment options, text in table cells is often aligned on a specified character, such as a decimal point. This character appears in the same horizontal location in all cells within a column. A column of prices is an obvious example, where the decimal point character is significant. The simplest way to ensure vertical alignment of the given character is to always place it exactly halfway across the cell, but this will not be appropriate if the cells are narrow and more text always appears to the right (as in a column of newspaper prices), or to the left (as in a column of house prices). The author could take account of the average or maximum amount of text that precedes or follows the special character in each of the cells in the column, or (at some risk) it could be left to the rendering application to calculate an appropriate position. The decimal points in the cells of the

final column in Figure 2 are closer to the right edge than to the left edge.

Straddling Cells

Headings often apply to several consecutive columns, and side-headings often apply to several consecutive rows, though any cell in a table should be able to straddle both columns and rows. In Figure 1, the first heading straddles two rows, the second straddles two columns, and one of the body cells also straddles two rows.

Straddling cells are not in themselves a great problem for rendering applications, but they increase the difficulty of calculating column widths, because it becomes much harder to establish how much text occurs in each column. Only some of the text in a horizontally straddling cell needs to be squeezed into the space that will be reserved for the first occupied column. There is no simple way to decide how much of this text should be "assigned" to the other columns, and in what proportions.

However, straddling cells can have a profound effect on the underlying XML markup. Table model designers have to decide whether or not an occupied cell should be represented

widgets with changed prices	prices	
	currency	amount
X123*	£	12
Y234	$	13.04
Z987*		8365.12
*These are shortly to be discontinued		

Figure 1 • Complex table authoring

widgets with changed prices	prices	
	currency	amount
X123*	£	12
Y234	$	13.04
*These are shortly to be discontinued		

widgets with changed prices	prices	
	currency	amount
Z987*		8365.12
*These are shortly to be discontinued		

Figure 2 • Complex table (split over pages)

by an empty XML cell element, or by nothing at all. In the latter case, it is necessary to study the rows above in order to be certain which column the content of a given cell element will be assigned to. Even the first cell element in a row does not necessarily contain text that will be placed in the first column. Authoring tools that can present the table grid, reveal the location and size of straddling cells, and allow cell straddling to be performed with immediate visual feedback (as shown in Figure 1), help to avoid otherwise inevitable authoring errors.

Complex Cell Contents

Often, the content of a cell will be a single block of text (a code, price, word, phrase, or paragraph). But a cell may need to be able to contain a narrative flow that includes any number and combination of headings, paragraphs, lists, and images. The most complex circumstance to arise is the need for a table within this flow, or even for a single table to be the only thing in the cell. When a rendering application treats a cell as simply another text area (that just happens to be smaller than the usual page or column areas), then it may not be concerned by such eventualities. But some applications do

> ## "…a cell may need to be able to contain a narrative flow that includes any number and combination of headings, paragraphs, lists, and images"

not work in this way, and have limited cell content formatting capabilities. As might be expected, the most common limitation is the inability of the application to render tables within a cell.

A safer but more limited solution is to use straddling cells to create the illusion of an embedded table. Consider a three-column, three-row table that must have a two-column, two-row table embedded in the middle cell. This can be simulated by actually making the main table four columns wide and four rows deep, so that the middle cell can become four separate cells. The audience is then fooled into thinking that the outer table has fewer columns and rows by straddling all other cells in the second column across to the third column, and stretching all other cells in the second row down into the third row.

Headers and Footers

When a large table occupies multiple pages of a printed document, it is common practice for the rendering application to repeat the header rows at the top of each of these pages. It is important to explicitly identify a header row, because it is not always just the first row. For example, a major heading that straddles several columns may lie over subheadings for each column (as shown in Figure 2).

Footnotes raise a similar problem, in that a footnote should be repeated at the base of each page-sized fragment of the table that includes a reference to that footnote (there are references to the same footnote on both pages in Figure 2), or more simply, at the bottom of every page that contains any

part of the table. If the reference precedes the footnote text in the XML data stream (as it does in Figure 1), then a page-break may occur before the rendering software has encountered the footnote text, and it must look ahead to find it.

Note that identification of header and footnote rows would also allow a browser to display the body rows within a scrollable area between fixed header and footnote rows.

Border Lines

Border lines are usually straightforward enough, despite the fact that a rendering application must reserve space for them, avoid drawing lines through the middle of straddling cells, and possibly avoid creating double lines when the table also contains a surrounding box. The only real problem that can occur is that an author attempts to use different border line styles to group columns, rows, or blocks of cells, when the rendering application is only able to draw one kind of border line. For this reason, even if the table model allows it, authors should be discouraged from using this grouping technique (empty rows or columns can be an effective alternative approach).

The Need for Standards

Document model designers could invent their own table models, introducing any number of relevant complex concepts, including some or all of the issues discussed above. But rendering applications could not be expected to understand the role of each element and attribute, and, even more seriously, could not be expected to deal with arbitrary complexities. Both of these problems would be solved if document model designers were to adopt an existing, well-known table model. Rendering applications could then be developed specifically to understand this model and to cope with all of its features.

There is an example document in the SGML standard that includes a table structure that is fully explained in a later Technical Report by the ISO (ISO/IEC TR 9573), but if it was ever intended to become a standard model, it failed dismally. In its place, two de facto table model standards have emerged. They are both discussed below, but note that there is insufficient space in this article to describe either in full, so the focus is on their strengths and weaknesses, particularly with respect to the issues previously discussed.

The CALS Standard

It was perhaps inevitable that a de facto table model would arise out of the most prominent early application of SGML – the interchange of documentation between the US Department of Defense and its subcontractors. The CALS (Continuous Acquisition and Lifecycle Support) standard for such documentation took the form of an SGML DTD, which naturally included a model for tables. Software packages that competed to support the CALS DTD were flexible enough to cope with the rendering of documents that conformed to other, arbitrary DTDs, but not to cope with other table models. So DTD authors were more or less obliged to adopt the CALS table model in their own DTDs. Its adoption in the DocBook standard (www.docbook.org), in particular, reinforced its status. XML versions of both CALS-related and DocBook DTDs have ensured the survival of this model.

From the start, it was expected that CALS documentation would be published on both paper and CD-ROM (there is little space for large paper manuals on a submarine), and the CALS table model therefore encourages the use of proportional column width settings. The values "1*", "3*", and "2*", for example, indicate that the second column must be three times the

BEA

dev2dev.bea.com/useworkshop

width of the first, and that the third must be double the width of the first. This is more convenient than using percentages because the author does not have to ensure that the values add up to a specific total (such as 100). Also, finer distinctions between column widths can be made by simply using larger values, so "50*" and "51*" indicate two columns where one is only very slightly wider than the other.

"While CALS remains preeminent in the SGML world, it has serious competition in the XML world"

The footnote problem is dealt with by forcing authors to create the footnotes in special footnote rows, before the body rows. The footnotes are then easy to place at the bottom of each page because there is no need for the rendering application to parse the table twice, or to look ahead, in order to first collect the footnote text:

```
<table>
  <thead>...</thead>
  <tfoot>...</tfoot>
  <tbody>...</tbody>
</table>
```

Each of these three partitions can contain rows and cells:

```
<tbody>
  <row><entry>cell 1,1</entry>
    <entry>cell 1,2</entry></row>
  <row><entry>cell 2,1</entry>
    <entry>cell 2,2</entry></row>
</tbody>
```

An entire table within a table cell is explicitly catered for, using an EntryTbl element instead of an Entry element, but some rendering applications may fail to support this feature.

The Char attribute specifies the character to vertically align columns of values on, and the CharOff attribute holds a value that represents a percentage of the column width from the left edge, indicating where to place this character.

The XHTML Standard

The CALS table model is both sophisticated and well established, yet another table model has risen to prominence in recent years. While CALS remains preeminent in the SGML world, it has serious competition in the XML world.

The HTML/XHTML table model is now at least as popular as CALS. This happened because XML has been influenced more by the HTML community than by the SGML community, despite the fact that XML is more closely related to SGML than to HTML. Although the first version of HTML was influenced by SGML, its inventors did not feel the need to adopt any of the existing SGML document models, simply because

HTML was aimed at new audiences and at new rendering tools (Web browsers), and specifically not at the demands of paper-based publishing. A simple table model was devised (by Netscape) then incorporated into HTML 3.2 (by the W3C).

The TR (Table Row) element holds all the cells of a single row, and the TH (Table Header) and TD (Table Data) elements both represent cells, with the former kind highlighted to show their importance as headings:

```
<table>
  <tr><th>header 1</th><th>header 2</th></tr>
  <tr><td>cell 1</td><td>cell 2</td></tr>
  <tr><td>cell 1</td><td>cell 2</td></tr>
</table>
```

Note that identification of header cells that can occur anywhere in a table is definitely no substitute for the identification of heading table rows that may need to be repeated at the top of each printed page.

As well as being able to contain text, a cell may contain any combination of any of the HTML formatting elements, including the Table element. Browsers tend to handle this possibility quite well, but other rendering applications may not do so.

As HTML matured, the table model acquired advanced CALS-derived features. It now optionally includes the THead, TFoot, and TBody elements to identify header and footnote rows. The Char attribute contains the character to be used to vertically align values in a column, and the Offset attribute specifies how far across the cell to place this character. The empty Col element at the top of a table can specify various characteristics of a single column, including its text alignment and width (although the most popular browsers recognize the Col element and cope with percentage values in the Width attribute, the more useful proportional "*" mechanism is not yet universally supported).

The popularity of the XHTML table model rests upon the fact that a number of other XML applications have adopted XHTML element names and element structures, including the table elements. This was the easy thing to do, because existing authoring tools could be used, and familiarity with the HTML model could be exploited to help ensure the acceptance of these DTDs.

Which Standard Model?

The HTML/XHTML model is now almost as sophisticated as the CALS model. Premium XML authoring tools tend to support both and allow WYSIWYG authoring and editing of tables in either form.

Regardless of which is chosen, if a rendering application requires empty elements to be present in grid locations occupied by straddling cells, then post-processing is needed to calculate where to insert these tags.

If the XHTML model is chosen, and the table authoring software, using pixels values, inserts cell widths directly into the TH and TD elements, then it may be necessary to get the author to add Col elements with fixed or percentage values (if the package allows), then either ignore or remove the individual cell widths.

Generated Tables

There are times when neither standard should be used, simply because it's not always a good idea to use table markup in a document instance. After all, a TD or Entry element says nothing about the information that it contains apart from

where to place it on the screen or page. If a cell contains a price, and there is a need to find, analyze, process, or extract prices, then it would be better if a Price element could be used instead. While the Price element could be placed within the TD or Entry element, this approach makes documents larger, and harder to write.

Structures might be created that look nothing like tables, though it helps if the information is organized in the correct order, with all of the data for one row occurring before the data for the next:

```
<stock>
  <item>
    <code>...</code><price>...</price><color>...</color>
  </item>
  <item>
    <code>...</code><price>...</price><color>...</color>
  </item>
</stock>
```

If this information then needs to be rendered in tabular form, it can be prepared by one of two possible approaches. XSLT can be used to convert the XML document into various formatting languages, including RTF, CALS, XHTML, and XSL-FO, which is very flexible, but requires document processing to occur before rendering. Alternatively, CSS (Cascading Style Sheets) rules can be applied to assign table-related roles to the elements, though the source data must be arranged in the required structure (because CSS cannot rearrange this data) and must be quite simple (although borders can be specified, header and footnote rows can be identified, and text can be vertically aligned, straddling can only be achieved, if at all, by

including the HTML attributes in the cell-level elements):

```
stock { display: table ; border: outset 5pt}
item  { display: table-row }
code  { display: table-cell }
price { display: table-cell }
color { display: table-cell }
```

The final problem with the CSS approach is that rendering applications often do not support these advanced CSS features in full.

Conclusions

Tables can be trickier than they first look. If simple table rendering is required, using XSL-FO or CSS to render a table from semantically marked-up content is an approach to consider, but otherwise the XHTML table model is now probably the safest bet, especially if Web browsers are one of the primary publishing channels. But consider disabling advanced features that may be unsupported in some rendering applications, and beware of column widths and different approaches to the markup of straddling cells. While tables can certainly be troublesome, with careful forethought they can also be tamed.

AUTHOR BIO

Neil Bradley is an XML consultant working for Rubus, a UK-based interactive systems integrator. Neil has specialized in publishing applications of SGML since 1986 and XML since its release, as a developer, analyst, and consultant. He is a regular speaker at related conferences and has written a number of related books: The Concise SGML Companion, The XML Companion, *and* The XSL Companion.

NEIL.BRADLEY@RUBUS.COM

Ektron

www.ektron.com/xml

XML, Ontologies, and the Semantic Web

The second generation of the Web

WRITTEN BY
AYESHA MALIK

"**I**f [computer networking] were a traditional science, Berners-Lee would win a Nobel Prize," Eric Schmidt, CEO of Novell, once commented. Indeed, Tim Berners-Lee revolutionized the world when he created the Web in 1991. Now, he is talking about the second generation of the Web, and his talks are generating buzz…the W3C is establishing standards for it, and universities, companies, and industry consortiums are building the technologies necessary for it. He refers to it as the Semantic Web.**

The Semantic Web is envisaged as a place where data can be shared and processed by automated tools as well as by people. The key lies in the automation and integration of processes through machine-readable languages. In order to leverage and link the vast amounts of information available on the Web, software agents must be able to comprehend the information, i.e., the data must be written in machine-readable semantics. For example, whether I use the tag <dead> or the tag <alive> next to a person's name in my XML document makes no difference to the parser. Some additional semantics or metadata must be added in order for a software program to make an intelligent assessment of the state of the person. This metadata, or meaning (versus display), of information is what is known as semantics.

Let's consider an example illustrating the advantages of having semantics that add meaning to information on the Web. Say you live in New York and decide to attend a conference in London. You would have to go to many airline Web sites and look at all flights leaving from New York to London. Then, you would go to various hotel Web sites and look for a hotel near your conference location that has a room available. That's a fair bit of searching. Luckily, you can search for the information on the Web, and in most cases you can pay for

everything on the Web.

Now imagine another scenario: you're driving down 5th Avenue in Manhattan. Your secretary calls you on your cell phone and says that you've been invited to be the keynote speaker at a conference in Europe on May 5, 2003. You think that's great, and you begin to make plans for your trip. You flip open your Palm Pilot, which is connected to the Web, and you type in some commands: book flight and return from New York to London, May 5–11; book room in hotel near the conference location, Hilton London Metropole, in London.

Your Palm Pilot has a software program or software agent that understands your commands; it processes the semantics of your command intelligently. Your agent buys your ticket and books a room in a hotel. As you drive into your garage, your Palm Pilot beeps and asks you to confirm the information. You park your car, confirm the bookings, and then go inside. This is just one example of how easy life gets when the Web is an intelligent partner in your universe.

Ontologies for Knowledge Representation

In order for computers to be more helpful, the Semantic Web augments the current Web with formalized knowledge and data that can be processed by computers. To be able to search and process information such as airline flights, software programs need information that has been modeled in a coherent manner. An ontology models all the entities and relationships in a domain.

Continuing with our example, let's create a hypothetical ontology for Virgin Atlantic's flights. An ontology for the airline industry would model its metadata using the following semantics (in italics):

A *flight* has an *origin, destination, flight number,*

departure time, arrival time, class {attributes}
A *international flight* is a type of *flight* {inheritance}
A *flight* can have *one origin* {one-to-one association}
A *flight* can have *many classes* {one-to-many association}

In other words, ontology captures the attributes of an entity and inheritance relationships as in object-oriented programming; it also captures associations such as cardinality as in relational databases (see Figure 1).

The specific information or instance of this metadata for a particular flight may be as follows:

Flight Number: VS018
Origin: New York (EWR)
Destination: London (LHR)
Departure Time: 08:20, May 5, 2003
Arrival Time: 20:00, May 5, 2003
Class: Economy

With these semantics, you can type the following commands for your software agent:

flight origin: "New York" *destination*: "London"
departure: "May 5, 2003" *arrival*: "May 5, 2003"

Without a standard naming convention for concepts such as destination, your software agent cannot present your commands to Virgin Atlantic's server. In addition, it is important that British Airways' server understands these semantics as well so that you can search for tickets on that airline. When you model the concepts in a domain, such as the airline industry, and publish them, you are in essence creating an ontology.

The Semantic Web Architecture

Now that we've discussed both the vision of the Semantic Web and the necessity of ontologies for knowledge representation, we'll explore the implementation of the model.

There are several important steps in the workflow of the example we discussed above:

1. Modeling the specifics of a resource such as Virgin Atlantic flight VS018 from New York to London. For this, we will discuss Resource Description Framework (RDF).
2. Modeling the concepts of the entire airline industry. Here we'll consider Web Ontology Language (OWL) and how to map one ontology to another.
3. Trusting that the information provided by an airline or a ticket broker is correct. We'll discuss digital signatures as well as an application for a trusted community known as Friend-of-a-Friend (FOAF).
4. The first three points consider information and its validity, but what about the mechanics of sending commands and receiving results? This involves a discussion of software agents and Semantic Web services, an extension of Web services.

An excellent starting point for any discussion of the architecture of the Semantic Web is Tim Berners-Lee's diagram shown in Figure 2. Discussing the different layers of the diagram will take us through the implementation of our example.

Unicode and Uniform Resource Identifier

The Uniform Resource Identifier (URI) forms the foundation of the Semantic Web. The URI provides a unique identifier for any Web resource, and even for any object outside the Web; for example, a person can have a URI. The Semantic

Web names every concept by a URI, thus letting anyone express new concepts with minimal effort and allowing definitions to be qualified by their sources.

Keeping Unicode as a foundation allows for the multiplicity of languages in which information is marked up throughout the globe. Unicode supports multilingual characters in a 40,000 character charset.

XML, Namespaces, and XML Schemas

Due to its flexibility, ability to be manipulated programmatically, and expressive power, XML (along with its associated technologies such as namespaces and schemas) is the most suitable language for a semantic language. The first contribution XML made to the Web was to separate content from representation; in the next iteration, XML is used to add metadata or meaning to content. Currently, the W3C is working on two main XML-based standards for the Semantic Web: Resource Description Framework (RDF) and Web Ontology Language (OWL). Once these standards become fully functional, parts of the Semantic Web should start to come together.
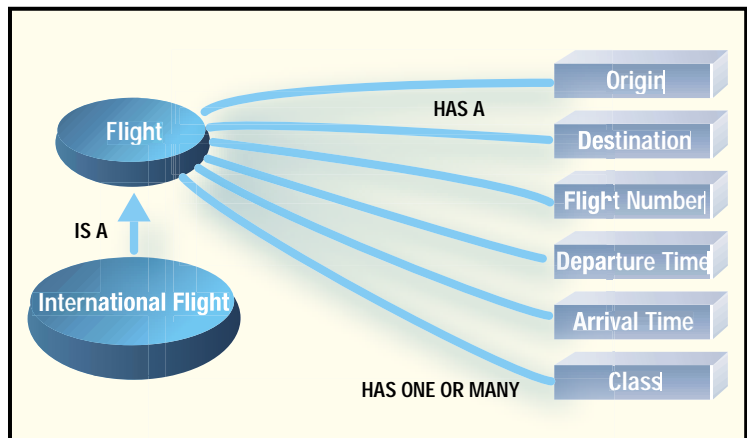

Figure 1 • Modeling an ontology for the airline industry

Resource Description Framework

RDF is one of the cornerstones of the efforts made in the direction of the Semantic Web. It is a language for representing information about resources in the World Wide Web and its syntax is XML. RDF represents a data model or metadata, i.e., a common framework for expressing information that can be shared across applications. According to the W3C, RDF represents information "by generalizing the concept of a 'Web resource.'"

The RDF framework is built on three pillars:

1. ***Resource:*** Anything that can have a URI; this includes all the Web's pages, as well as individual elements of an XML document. An example of a resource is http://www.example.org/flight.
2. ***Property:*** A resource that has a name and can be used as a property, for example, Origin or Destination.
3. ***Statement:*** Comprises the data model for RDF and consists of the combination of a resource, a property, and a value. For example, if the resource is "VS018", the property is "Origin", and the Value in this statement is "New York".

Virgin Airlines stores the information about flight VS018 from New York to London in XML-based RDF in the manner shown in Listing 1. The RDF in Listing 1 describes our flight from New York to London very accurately, but what if you're going to another conference in San Diego in June and you want to search for a flight to San Diego? Software agents
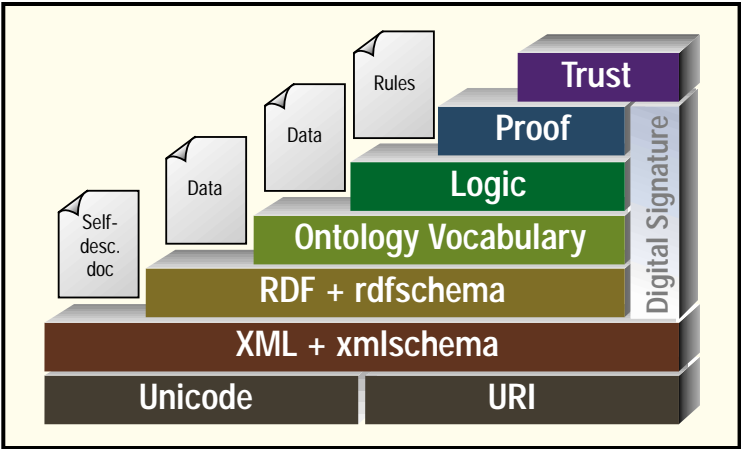
Figure 2 • The Semantic Web architecture (Source: Tim Berners-Lee)

require that similar concepts be described in the same manner in order to search information efficiently. In other words, each industry has to design the metamodel of the information pertinent to its domain. This calls for a schema to constrain and formalize the language of the RDF, i.e., to specify what constitutes a generic "flight." For the Semantic Web, we can use RDF schemas, or even better, we can use the OWL to model ontologies written in RDF.

Web Ontology Language

OWL is the XML Schema for RDF; OWL allows the definition of new vocabularies and ontologies that are written in RDF. According to the W3C, OWL is "intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications." Just as RDF has triplets of subject, predicate, and object, OWL has classes and properties and constraints on the way those classes and properties can be employed. A set of OWL assertions loaded into a reasoning system is called a knowledge base (KB). OWL is used to publish and share ontologies on the Web.

Let's look at our example and discuss inheritance, cardinality, and association. One example of a class is Flight. A Flight class has properties such as Number, Origin, Destination, Departure, Arrival, and Class.

Namespaces

Namespaces are used at the top of the .owl file to specify the origin of the various vocabularies used in the document.

```
<rdf:RDF
 xmlns      = "http://www.example.org/flight#"
 xmlns:owl = "http://www.w3.org/2002/7/owl#"
 xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs= "http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd = "http://www.w3.org/2000/10/XMLSchema#">
```

Once namespaces are established we begin with an assertion that what follows is an OWL ontology.

```
<owl:Ontology rdf:about="http://www.example.org/flight">
```

Classes

The most basic concepts in a domain should correspond to classes that are the roots of various taxonomic trees. Every individual in the OWL world is a member of owl:Thing. Thus each user-defined class is implicitly a subclass of owl:Thing.

Domain-specific root classes are defined by simply declaring a named class. For our sample contacts domain, we create one root class: Flight.

```
<owl:Class rdf:ID="Flight"/>
```

Inheritance

Say we want to divide our flights into international and domestic flights. Hence, InternationalFlight will be a subclass of Flight. We express this in OWL in the following manner:

```
<owl:Class rdf:ID="InternationalFlight">
  <rdfs:subClassOf rdf:resource="#Flight" />
  ...
</owl:Class>
```

Properties

Properties are used to make assertions about classes. There are two types of properties: object and datatype. Properties can express constraints between elements. Two kinds of restrictions that can be used are the domain of a property and the range of a property.

```
<owl:ObjectProperty rdf:ID="flightsPerDay">
  <rdfs:domain rdf:resource="#InternationalFlight"/>
  <rdfs:range rdf:resource="#Integer"/>
</owl:ObjectProperty>
```

We're defining a property that specifies how many flights per day occur for an international flight, and the property is restricted to an Integer.

Cardinality

It's possible to specify the type of association in terms of cardinality between two entities. In our example, we note that there is a restriction on the property flightsPerDay for an international flight, which says that there is only one international flight per day. This is important information for a software agent that is searching for options on which flight to get for the New York to London leg.

```
<owl:Class rdf:ID="InternationalFlight">
  <rdfs:subClassOf rdf:resource="#Flight"/>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#flightsPerDay"/>
      <owl:minCardinality>1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</Class>
```

Ontology Mapping

The key to ontologies is that they can be shared and therefore increase efficiency and interoperability. However, it is sometimes the case that two different organizations have two different names for the same concept, i.e., the ontologies are different. In such cases, the ability to map RDF schemas or ontologies is crucial to maintaining the advantages of the Semantic Web.

In OWL, there are several constructs that can be used for ontology mapping. Two of these attributes are sameClassAs and samePropertyAs – they indicate that a particular class or property in one ontology is equivalent to a class or property in

another ontology.

Say, for example, that British Airways calls a flight by the class name "AirJourney" while Delta refers it as "Flight". How can a software agent know that the two mean the same thing (and recall that semantics are about concepts and meanings)? Ontology mapping between the British Airways and Delta ontologies is required. An example is shown below.

```
<owl:Class rdf:ID="Flight">
  <owl:sameClassAs rdf:resource="AirJourney"/>
</owl:Class>
```

Web of Trust

We can model the information, but how do we trust the information that we get from the Semantic Web, and how do we protect our information? If my software agent finds two travel agents, and one says the price for a Virgin Atlantic ticket is $180 while the other says the price is $210, whom do I believe? In the Semantic Web, we depend on digital signatures and community networks.

Digital signatures

Digital signatures are necessary to ensure that the information that claims to be coming from a source was not tampered with before it got to you, and that its origin was indeed the named source. Based on mathematics and principles of cryptography, digital signatures allow signed RDF documents to be trusted. According to the W3C, "The combination of metadata and digital signature capabilities will aid in building a genuine Web of Trust."

Digital signatures address the problems of message integrity, data origin authentication, signer authentication, and non-repudiation of sending a message. Furthermore, signed XML combined with the RDF will provide a layer of authentic metadata that will improve search engine capability, support intelligent software agents, and create new ways of cataloging information for improved navigation.

FOAF (Friend of a Friend)

Even if we could verify that the information did come from a particular source, how would we decide whether to trust that source? One way is to trust sites that have been verified as trustworthy by organizations or even a community of your friends. The latter thought was the impetus behind the Friend-of-a-Friend (FOAF) idea.

FOAF falls under the rubric of social networking. The FOAF vocabulary allows you to specify the information necessary for membership to a community, such as name and e-mail address. However, you could augment this information to find out about the interests of other members, or even, in line with our argument above, to gather information regarding which site to trust.

FOAF provides one opportunity to build a prototype of the "Web of Trust" that Berners-Lee refers to in his Semantic Web roadmap.

Software Agents and Semantic Web Services

Ontologies comprise the knowledge representation component of the Semantic Web, but it is incomplete without software programs that can communicate with each other. We still need a mechanism by which a software agent goes to Virgin Atlantic and requests information on flights to London from New York on May 5, 2003.

The best application for invoking other applications on the Web using request parameters is Web services. Web services is a layer of abstraction above software programs and allows services to be located and invoked across the Web. Programs written in various programming languages on different platforms can call each other using the Web services interface. The services offered by a company are published in a public registry that's used to locate them. Method invocations and results are communicated using the Web services messaging framework. Web services is important for universal interoperability and integration and will be the key enabler for software agents in the Semantic Web.

There are three main standards for Web services: WSDL (Web Service Description Language), an XML-based syntax that describes the functions provided by a Web service; UDDI (Universal Description, Discovery, and Integration), an XML-based syntax used to develop a registry of services that can be published on the Web; and SOAP (Simple Object Access Protocol), the most common protocol for carrying the messages that invoke Web services. Using Web services, Virgin Atlantic can publish its services in a registry and anyone who wants to call a find flight command can send a message to invoke the method published in its registry.

DARPA (Defense Advance Research Projects Agency) has been working on an extension of Web services known as Semantic Web services. Semantic Web services have a declarative, machine-readable API for services. The API would inform the agent of how to use the service, which parameters to provide, and what results will be returned. So Semantic Web services are an enhanced version of Web services; they formalize the language in which we describe and call Web services. DARPA has developed DAML-S, an ontology or semantics for describing the properties and capabilities of Web services. DAML-S sits at the application level above WSDL and describes what is being sent as opposed to describing only how it is being sent (which is the functionality provided by WSDL). In other words, DAML-S will complement WSDL.

Using DAML-S, the user will not have to specify the Web service that it wants; the software agent will be able to discover the capability required by the consumer by looking at the declarations of the capabilities advertised by the marked up Web services. Next, it will compose tasks itself, i.e., it will both find the flight and buy the ticket for it (the composition of complex tasks is not possible with the current state of Web services). Semantic Web services would, therefore, greatly enhance the capability of software agents to search for particular services and are an important step in the direction of implementing the Semantic Web.

Present Efforts and Future Directions

The Semantic Web is the second-generation Web. It weaves together a network of information, which allows more efficiency, greater knowledge-sharing, and ease of use. Ontologies are the key to this interoperability because they determine the language software agents will need to communicate with each other and humans will need to communicate with the agents. As we have seen in this article, the semantics necessary for ontologies are written in XML, or more specifically RDF and OWL using XML syntax.

There are three factors necessary for the success of the Semantic Web: first, the establishment of standards by the W3C; second, the development of technologies that facilitate the implementation of software agents and other aspects of Berners-Lee's vision; and third, the production of tools that encourage people to adopt the technologies that will facilitate the universality of the Semantic Web.

The W3C, led by Tim Berners-Lee and Eric Miller, has made

great progress in the standards established for the Semantic Web. In 2002, several new recommendations and working drafts have emerged for OWL and RDF, the two main standards for the Semantic Web. Detailed examples and guides are provided for users who want to mark up their information on the Web.

Technologies such as Web services and digital signatures are just two examples of relatively recent developments that will greatly facilitate the implementation of the Semantic Web. Examples of implementations include Music Brainz (www.musicbrainz.org), which provides an encyclopedia of music marked up in RDF; Friend-Of-A-Friend (http://xmlns.com/foaf/0.1), which uses RDF to mark up the identity of community members and provides a basis for a Web of trust; and Retsina Calendar Software Agent (www.softagents.ri.cmu.edu/Cal), which is an agent developed for calendar scheduling by Carnegie-Mellon University.

Regarding encouraging people to mark up their Web information, I tend to agree with James Hendler, Professor at the University of Maryland and a prolific writer on the Semantic Web, that "ideally, most users shouldn't even need to know that Web semantics exist." Tools must be constructed that automatically pop up forms for ontology linkages in order to overcome the initial hesitation that people have in learning semantic markup languages. DARPA is funding a number of such free tools so people will mark up their Web pages. One example is an ontology editor, Protégé, developed by Stanford University, which is free and available for download from the Stanford Web site.

Of course, we have spoken of more than just individual Web pages; in our hypothetical example, we considered the importance of ontologies and these are usually developed by industry consortiums. Luckily, creating ontologies is something that is already underway. Many industries have realized that they need industry standards to facilitate inter- and intra-firm communication. One example of this is FpML (Financial Product Markup Language), an ontology for financial instruments written in XML syntax. Its goal is to establish a representation of concepts – an ontology – for all financial trading firms to be able to use for their trading purposes. If industry consortiums that are creating ontologies in XML instead write them in RDF, which is an XML-based syntax, they will have taken the first but important step toward creating the Semantic Web.

These efforts all point to the growing importance, and in my mind, the inevitability of the establishment of the Semantic Web. Just because it sounds like science fiction doesn't mean it's impossible. The Semantic Web is an incredibly exciting and potential place for developers to work. It will revolutionize the way we interact, live, and do business today. If you have seen movies like *The Matrix* and *Minority Report*, you've glimpsed the new kind of artificial intelligence that uses the Web to process information rapidly and automatically. Who knows? One day, you might very well be able

to just speak to your small Palm Pilot or laptop instead of typing in the commands. Even today, companies such as IBM produce simple voice recognition software that allows you to speak to your computer. The key is that the computer needs a defined set of semantics for it to understand your commands and be able to communicate with other software programs on the Web. For now, the W3C is defining standards, new technologies like Web services and XML Schemas have emerged that will make the transition easier, and industries and companies are focusing on making better models to represent their knowledge.

I predict that industries will develop ontologies that will be used for their internal communication. Eventually, each industry such as financial services, retail, and shipping will merge its internal ontologies and represent a coherent protocol for communication with its systems. At that point, the Semantic Web will evolve from existing in pockets to becoming a universal infrastructure. Eventually, with increasing unambiguous markup of Web content, the Semantic Web will evolve to Tim Berners-Lee's vision as "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

Resources

- Berners-Lee, Tim. (1998). "Semantic Web Roadmap": www.w3.org/DesignIssues/Semantic.html
- *Resource Description Framework:* www.w3.org/RDF/
- Dumbill, Ed. (2000). "The Semantic Web: A Primer": www.xml.com/pub/a/2000/11/01/semanticweb
- Palmer, Sean B. (2001). "The Semantic Web: An Introduction": http://infomesh.net/2001/swintro
- *Web Ontology Language (OWL) Guide (Version 1.0, W3C Working Draft 4 November 2002):* www.w3.org/TR/2002/WD-owl-guide-20021104
- *Semantic Web Activity: Advanced Development*: www.w3.org/2000/01/sw
- Cowles, Paul. (2002). "Web Services and the Semantic Web." *Web Services Journal*. December: www.sys-con.com/webservices/article.cfm?id=419
- Hendler, James. (2001). "Agents and the Semantic Web." IEEE *Intelligent System*. Number 2.
- Berners-Lee, Tim; Hendler, James; and Lassila, Ora. (2001). "The Semantic Web." *Scientific American*. May: www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21
- *DARPA Agent Markup Language:* www.daml.org
- "DAML-S: Web Service Description for the Semantic Web" by The DAML Services Coalition. The First International Semantic Web Conference (ISWC), Sardinia (Italy), June 2002: www.daml.org/services/ISWC2002-DAMLS.pdf
- McIlraith, S.; Son, T.C.; and Zeng, H. (2001). "Semantic Web Services." IEEE *Intelligent Systems*. March/April: www.daml.org/services/ieee01-KSL.pdf
- Dumbhill. Ed. (2002). "XML Watch: Finding Friends with XML and RDF." IBM developerWorks. June: www-106.ibm.com/developerworks/xml/library/x-foaf.html

AUTHOR BIO

Ayesha Malik is a senior consultant with Object Machines, a software engineering firm providing Java technology and XML solutions to businesses. She serves on the Architecture Working Group of Financial Products Markup Language (FpML), a data-interchange standard set forth by International Swaps and Derivatives Association (ISDA). Ayesha holds a BA with honors from Harvard University and an MS from Columbia University, where she studied operations research, applied mathematics, and computer science.

AYESHA.MALIK@OBJECTMACHINES.COM

> LISTING 1

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
             xmlns:f="http://www.example.org/flight">
   <rdf:Description
rdf:about="http://www.example.org/flight#VS018">
          <f:number>VSOI8<f:number>
          <f:origin>New York</f:origin>
          <f:destination>London</f:destination>
          <f:departure>08:20 05/05/2003</f:departure>
          <f:arrival>20:00 05/11/2003</f:arrival>
          <f:class>Economy</f:class>
   </rdf:Description>
</rdf:RDF>
```

▼ Download the Code
▼ www.sys-con.com/xml

| Boston 2003 | London 2003 | Berlin 2003 | Hong Kong 2003 |

JDJ EDGE conference&expo

web services EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

Register by March 14th
SAVE UP TO **$200**

3rd Annual

Web Services Edge EAST

Conference & Exposition

FINAL PROGRAM

Connecting the Enterprise with Web Services, Java, XML, & .NET

March 18–20, 2003
Hynes Convention Center
Boston

CALL TODAY TO REGISTER
201-802-3058
OR REGISTER ONLINE AT
www.sys-con.com

Event Sponsors Microsoft · Sun microsystems · SYBASE · IONA · DIGITAL EVOLUTION The Business Value of Web Services
COMPUWARE · ORACLE · hp invent · Rational the e-development company · ALTOVA

Media Sponsors JAVA DEVELOPER'S JOURNAL · JavaWorld · WebServices JOURNAL · LINUX BUSINESS WEEK · XML JOURNAL · SAMS · OASIS · PerfectXML · web services.org
.NET DEVELOPER'S JOURNAL · SD Times · asp.netPRO · WebSphere DEVELOPER'S JOURNAL · WebLogic DEVELOPER'S JOURNAL · wireless BUSINESS & TECHNOLOGY · CF Advisor · COLDFUSION Developer's Journal · .net · HSP · PowerBuilder Developer's Journal

Special Insert: Web Services Edge East Conference & Expo
Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

web services EDGE conference&expo

JDJ EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

THE LARGEST WEB SERVICES CONFERENCE & EXPO
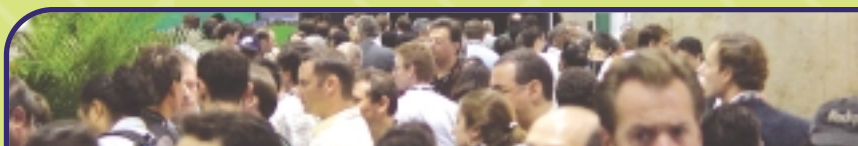
FEATURES & ATTRACTIONS

▶ **3 Days Packed with Education and Training**

▶ **4 Keynotes & 3 Panel Discussions**

▶ **60 Hard-Hitting and Current Seminars**

▶ **FREE .NET Tutorial with Russ' Tool Shed**

▶ **Web Services & XML Tutorials**

▶ **Java University Certification Training**

▶ **HOT** Industry-Leading Certification Programs

▶ **1 Day Mobile .NET Tutorial**

▶ **INFORMATIVE** Round Table Discussions

▶ Opening Day Welcome Reception

▶ SAMS Meet the Authors Hot Topics Lounge

▶ **COMPELLING** Case Studies & Best Practices

▶ **FEATURED** Product Demonstrations on the Show Floor

▶ **RIVETING** Real-time SYS-CON Radio Interviews

WHO SHOULD ATTEND

- Software Developer
- Software Engineer
- Development Manager
- Application Developer
- Technical Director
- Analyst/Programmer
- IT Manager
- Technical Architect
- Team Leader
- Software Consultant

Keynotes & Highlighted Speakers

John Magee
Vice President, Oracle9i

ORACLE

John Magee is vice president, Oracle9i, at Oracle. He has more than 14 years' experience in the enterprise software industry and has held positions in product development, product management, and product marketing. In his current role, Magee manages technical product marketing for Oracle's application server and development tools products, and is responsible for evangelizing Oracle technology initiatives around J2EE, XML, and Web services.

Mark Herring
Director, Java, Web Services & Tools Business

Sun microsystems

Mark Herring is responsible for helping to define, set, and drive Sun Microsystems' product direction in the Java, Web Services & Tools Business. Prior to his current role, Herring was director of corporate strategy & planning, looking after Sun's interest in the Project Liberty Alliance and Network Identity. Herring joined Sun Microsystems in October 1999 as a result of Sun's acquisition of Forte Software. Forte Software was a leading provider of enterprise-class development and integration products. During his four years at the company, he ran several aspects of Forte's marketing organization, including product marketing and the Web channel.
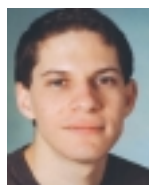
Miguel de Icaza
Cofounder and CTO

As the founder and leader of the GNOME Foundation, Miguel de Icaza is one of the foremost luminaries in the Linux development community. With his seemingly boundless energy, de Icaza has galvanized the effort to make Linux accessible and available to the average computer user. He brings this same excitement to his role as CTO of Ximian. de Icaza was instrumental in porting Linux to the SPARC architecture and led development of the Midnight Commander file manager and the Gnumeric spreadsheet. He is also a primary author of the design of the Bonobo component model, which leads the way in the development of large-scale applications in GNOME.

Mark Hapner
Distinguished Engineer, Sun Microsystems

Mark Hapner is a Sun Distinguished Engineer and is currently lead architect for Java™ 2 Platform, Enterprise Edition (J2EE™). He has guided the overall architecture for J2EE 1.2, 1.3, and now the upcoming 1.4 release. In March of 1996, he joined Sun's JavaSoftware Division to participate in the development of the Java database connectivity API (JDBC). Following that, he was co-spec lead of the Enterprise JavaBeans specification and spec lead of the Java Message Service specification.

Simon Phipps
Chief Technology Evangelist, Sun Microsystems

Simon Phipps, currently chief technology evangelist at Sun Microsystems, speaks frequently at industry conferences on the subject of technology trends and futures. He was previously involved in OSI standards in the 1980s, in the earliest collaborative conferencing software in the early 1990s, and in introducing Java and XML to IBM.

Dave Chappell
VP, Chief Technology Evangelist, Sonic Software

Dave Chappell is the vice president and chief technology evangelist for Sonic Software. He has more than 18 years of industry experience building software tools and infrastructure for application developers, spanning all aspects of R&D, sales, marketing, and support services. Chappell has published in numerous technical journals, and is currently writing a series of contributed articles for *Java Developer's Journal*.

Eric Newcomer
Chief Technology Officer, IONA

In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's Orbix E2A e-Business Platforms as relates to standards adoption, architecture, and product design. Newcomer joined IONA in November 1999, and most recently served as IONA's vice president of engineering, Web Services Integration Products. He is a member of the XML Protocols and Web Services Architecture working groups at the W3C and IONA's Advisory Committee representative to UDDI.org.

Dean Guida
CEO and President, Infragistics

Dean Guida is CEO and president of Infragistics and was CEO and a cofounder of ProtoView Development Corporation. Mr. Guida has over 15 years of experience in the technical industry and oversees all aspects of the company's business operations and corporate direction. He is also responsible for cultivating strategic alliances and other external relationships, as well as managing corporate financial affairs.

Conference Advisory Board

Sean Rhody
Editor in Chief,
Web Services Journal
Partner, CSC

Alan Williamson
Editor in Chief,
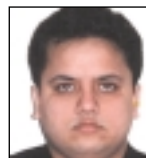*Java Developer's Journal*
Chief Technology
Officer, n-ary

Derek Ferguson
Editor in Chief,
.NET Developer's Journal
Chief Technology Evangelist,
Expand Beyond Corp.

Bob Familiar
.NET Architect,
Microsoft
New England

Thomas Robbins
Senior Technology Specialist,
Microsoft
New England

Hitesh Seth
Editor in Chief,
XML-Journal
Chief Technology Officer, ikigo

J.P. Morgenthal
Chief Services Architect,
Software AG

Register Online @ sys-con.com

Conference at-a-Glance

TUESDAY MARCH 18

Time	JAVA	WEB SERVICES	.NET
8:00AM – 4:00PM	Registration Open		
9:00AM – 9:50AM	(JV1) Squeezing the Best Out of Java Alan Williamson, Java Developer's Journal	(WS1) Web Services Infrastructure Carl Sjogreen, BEA	(NT1) .NET Framework Overview Bob Familiar, Microsoft
10:00AM – 10:50AM	Web Services Keynote: John Magee, Oracle		
11:00AM – 11:50AM	(JV2) Testing Your Java Using JUnit Kyle Gabhart, LearningPatterns	(WS2) Web Services Management James Phillips, Actional	(NT2) Introduction to ASP.NET Russ Fustino, Microsoft
1:00PM – 1:50PM	WS-I Panel: "A Road Map for Web Services Standards" - Moderated by Rob Cheng, WS-I		
2:00PM – 2:50PM	.NET Keynote: "The MONO Project" - Miguel de Icaza, Ximian		
3:00PM – 3:50PM	(JV3) Building/Deploying the Ant Way Kyle Gabhart, LearningPatterns	(WS3) Strategies for Using Databases in a World of Web Services Mike Lehmann, Oracle	(NT3) Introduction to VB.NET Russ Fustino, Microsoft
4:00PM – 4:50PM	(JV4) Unlocking the Secrets of JDK1.4 Raghavan Srinivas, Sun Microsystems	(WS4) Using Web Services to Integrate J2EE and .NET Enterprise Applications - Odysseas Pentakalos, SYSNET International	(NT4) How to Develop an End-to End .NET Connected Appilication Allan da Costa Pinto, Microsoft

WEDNESDAY MARCH 19

Time	JAVA	WEB SERVICES	.NET
8:00AM – 4:00PM	Registration Open		
9:00AM – 9:50AM	(JV5) Java APIs for Web Services Security Standards Sang Shin, Sun Microsystems	(WS5) Combining BPM and BRM Technologies: A Major Step Towards Corporate Agility Henry Bowers, ILOG	(NT5) .NET: The Virtualized Execution Engine Yahya Mirz, Aurora Borealis
10:00AM – 10:50AM	Java Keynote: Mark Herring, Sun Microsystems		
11:00AM – 6:00PM	EXPO OPEN 11:00 a.m. - 6:00 p.m.		
11:00AM – 11:50AM	(JV6) Using SWT Under Eclipse\ Alternative - IBM	(WS6) Web Services for Real-Time Data Access in an Industrial Setting Stephan Van Dijck, ABB/SKYVA	(NT6) Introduction to DotGNU Barry Fitzgerald, DotGNU
12:00PM – 2:00PM	BREAK & EXPO		
2:00PM – 2:50PM	.NET Panel Discussion - Moderated by Derek Ferguson, .NET Developer's Journal		
3:00PM – 3:50PM	(JV7) Unlocking the Power of XML Hitesh Seth, ikigo	(WS7) Web Services Architecture: The Next Big Spec. from the Mouths of the W3C Eric Newcomer, IONA (moderator)	(NT7) Introduction to SSCLI Yahya Mirz, Aurora Borealis
4:00PM – 4:50PM	(JV8) Java and .NET Derek Ferguson, Expand Beyond	(WS8) Web Services: Next Steps After the Hype Claire Dessaux, Oracle	(NT8) Mobile Development with the Compact Framework Brad McCabe, Infragistics

THURSDAY MARCH 20

Time	JAVA	WEB SERVICES	.NET
8:00AM – 4:00PM	Registration Open		
9:00AM – 9:50AM	(JV9) Writing SOAP Services Nigel Thomas, SpiritSoft	(WS9) Web Services Best Practices Chris Peltz, HP	(NT9) Best Practices for .NET Development Joe Stagner, Microsoft
10:00AM – 10:50AM	.NET Keynote - Jesse Liberty, Liberty Associates		
11:00AM – 4:00PM	EXPO OPEN 11:00 a.m. - 4:00 p.m.		
11:00AM – 11:50AM	(JV10) Working with Data the JDO Way Patrick Linsky, SolarMetric	(WS10) Web Services Startups: Telltales of the Future Simeon Simeonov, Polaris Venture Partners	(NT10) Best Practices for ADO.NET Development Thom Robbins, Microsoft
12:00PM – 2:00PM	BREAK & EXPO		
2:00PM – 2:50PM	Java Panel - "The Future of Java", Moderated by Alan Williamson, Java Developer's Journal		
3:00PM – 3:50PM	(JV11) Enterprise: The Next Generation Mark Hapner, Sun Microsystems	(WS11) Open Standards for Web Services Messaging Dave Chappell, Sonic Software	(NT11) How to Debug with .NET Tony Denbow, STAR Information Technology
4:00PM – 4:50PM	(JV12) Overcoming the Challenges of J2ME Dr. Jeff Capone, Aligo	(WS12) Web Services Security Marc Chanliau, Netegrity	(NT12) XML and Web-Enabling Legacy Applications Using BizTalk Mike Cramer, Microsoft

XML	VENDOR	JAVA UNIVERSITY℠ PROGRAM	FAST TRACKS & TUTORIALS
(XM1) XML - A Manager's Guide JP Morgenthal, Software AG	Visit www.sys-con.com for details		
(XM2) OASIS Standards Update Karl Best, OASIS	(VN2) The XMLSPY 5 Enterprise Edition Development Environment Trace Galloway, Altova	**Sun** microsystems	**XML EDGE** conference & expo
		9:00AM – 5:00PM **Web Services Programming Using Java™ Technology and XML** This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.	9:00AM – 5:00PM **XML Certified Developer Fast Path** This tutorial is for programmers who have some knowledge of XML and related technologies and would like to pass the IBM Certified Developer Test 141 on XML and Related Technologies.
(XM3) A Definitive Introduction to XML Schemas Hitesh Seth, ikigo	(VN3) SOAP and Java: Marrying Them Off Skip Marler, Parasoft		
(XM4) XML in Print - XSL:FO Frank Neugebauer, IBM	Visit www.sys-con.com for details		
(XM5) XML Security Integration Challenges Phil Steitz, American Express	(VN5) Process-Centric Enterprises Eric Pulier, Digital Evolution	**Sun** microsystems	**Microsoft**®
		9:00AM – 5:00PM **Java 2 Platform Programmer Certification Fast Path** This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.	9:00AM – 5:00PM **Russ' Tool Shed** Join Russ as he shows you how to use Visual Studio.NET. 9:00-12:15 - Introduction to Web Services Using VS.NET 1:00-2:30 - Advanced Web Services Using ASP.NET 2:45-4:15 - .NET Remoting Essentials **RUSS' TOOLSHED** TINKERING WITH VISUAL STUDIO
(XM6) Case Study: XML in Life Sciences Tim Matthews, Ipedo	(VN6) Pattern Driven Application Development Tom Shore, Compuware		
(XM7) Using XML for EAI - Best Practices Dan Enache, TIBCO	(VN7) Managing the Developer Relationship Mike Bellissimo, Sun Microsystems		
(XM8) Delivering Web Services to Mobile Clients with XML Transformation Frameworks Paul Lipton, Computer Associates	(VN8) Web Services Diagnostics Dave Seidel, Mindreef		
(XM9) Delivering Web Services to Mobile clients with XML Transformation Frameworks - Paul Lipton, computer Associates	Visit www.sys-con.com for details	**Sun** microsystems	Microsoft® **.net**
		9:00AM – 5:00PM **Java 2 Platform Architect Certification Fast Path** This intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives.	9:00AM – 5:00PM **Mobile .NET** In this session, Derek Ferguson, editor-in-chief of .NET Developer's Journal, will give you a thorough introduction to the use of .NET with all manner of mobile computing devices.
(XM10) XQuery Mike Champion, Software AG	(VN10) Model Driven Development of Web Services in UML for the J2ME Bill Graham, Rational Software		
(XM11) XPath & XSLT 2.0 Kurt Cagle, Cagle Communications	(VN11) Why Web Services Management? Jon Atkins, HP		
(XM12) Third Generation XML Tools Michael Leventhal, Commerce One	Visit www.sys-con.com for details		

Java Technology Track

The Java track has been specifically designed to allow you to squeeze as much information out of each session as possible. This track is designed for the Java developer who wishes to catch up on the latest techniques and APIs and will be led by industry-leading speakers and authors.

The Java track has been designed with you, the more experienced Java developer, in mind. We know you don't have a lot of spare time, and we've designed the track to ensure that your time is maximized and you are armed with all the necessary tools to take your development to the next level.

(JV1) Squeezing the Best Out of Java
ALAN WILLIAMSON, JAVA DEVELOPER'S JOURNAL

Tuesday March 18, 2003 9:00 A.M. - 9:50 A.M.

Java is a very powerful language, and while it offers the developer a rich array of tools, the fundamentals should not be overlooked. Improving your code at the core layer will result in great improvements in efficiency and fewer bugs. We'll look at the dos and don'ts of programming and learn lots of hints and tips that will accelerate your Java coding.

BIO: Alan Williamson is editor-in-chief of *Java Developer's Journal*. In his spare time, he holds the post of chief technical officer @ n-ary (consulting) Ltd, one of the first companies in the UK to specialize in Java at the server side. Reach him at alan@n-ary.com (http://www.n-ary.com). Rumor has it he welcomes all suggestions and comments!

(JV2) Testing Your Java the JUnit Way
KYLE GABHART, LEARNINGPATTERNS

Tuesday March 18, 2003 11:00 A.M. - 11:50 A.M.

A critical measure of the success of software can be found in whether or not it executes successfully. Equally important, however, is whether or not that software does what it was intended to do. JUnit is an open-source testing framework that provides a simple means for developers to define their intentions for how their software should work. JUnit then provides test runners that process your intentions and verify that your code performs as intended. The result is software that not only works, but works in the correct way.

BIO: Kyle Gabhart is a senior mentor for LearningPatterns, a dynamic knowledge company providing consulting, training, and mentoring in emerging technologies. He is a prolific writer, with more than two dozen technical articles and books to his name. Kyle is highly regarded as a dynamic and enthusiastic public speaker with an innovative perspective on technology.

(JV3) Building/Deploying the Ant Way
KYLE GABHART, LEARNINGPATTERNS

Tuesday March 18, 2003 3:00 P.M. - 3:50 P.M.

A defined and easily repeatable process is one of the most necessary but often least-used aspects of good software development. A defined build process ensures that your project's software is built, deployed, and tested identically each time. Without this type of control and predictability, valuable time is often lost chasing down bugs that don't exist or rejecting solutions that were only partially implemented.

Apache's Ant is a powerful scripting tool that enables developers to define and execute routine software development tasks using the simplicity and extensibility of XML. Ant provides a comprehensive mechanism for managing software development projects, including compilation, deployment, testing and execution. Additionally, it is compatible with any IDE or operating system.

BIO: Kyle Gabhart is a senior mentor for LearningPatterns, a dynamic knowledge company providing consulting, training, and mentoring in emerging technologies. He is a prolific writer, with more than two dozen technical articles and books to his name. Kyle is highly regarded as a dynamic and enthusiastic public speaker with an innovative perspective on technology.

(JV4) Unlocking the Secrets of JDK 1.4
RAGHAVAN SRINIVAS, SUN MICROSYSTEMS

Tuesday March 18, 2003 4:00 P.M. - 4:50 P.M.

With the release of JDK 1.4, a number of new features were added to the core API, such as nonblocking IO, regular expressions, assertions, and XML. This session will take you through the major additions and demonstrate where you might use them.

BIO: Raghavan Srinivas is a Java technology evangelist at Sun Microsystems who specializes in Java and distributed systems. He has spoken on a variety of technical topics at conferences around the world, and brings with him more than 15 years of software development experience. Prior to joining Sun, Raghavan worked for Digital Equipment Corporation. He has worked in several technology areas, including internals of VMS, UNIX, and NT.

(JV5) Java APIs for Web Services Security Standards
SANG SHIN, JAVA TECHNOLOGY EVANGELIST, SUN MICROSYSTEMS

Wednesday March 19, 2003 9:00 A.M. - 9:50 A.M.

Everybody is talking about Web services as a way to perform business transactions over the Web in ways never done before. Yet, security is the most critical piece that still needs to be addressed before the promise of Web services can be realized. This session introduces the various Web services security standards, such as XML signature, XML encryption, XKMS (XML Key Management Services), XACML (eXtensible Access Control Markup Language), SAML (Security Assertion Markup Language), WS-Security, and Liberty First and their corresponding Java APIs, especially the standards APIs that are currently being defined through the Java Community Process (JCP). Wherever possible, example code will be presented.

BIO: Sang Shin has been with Sun Microsystems for over 12 years, working in various research and engineering projects mostly in data communication, networking, Internet, and Java-related areas. Prior to Sun, he worked for several startup companies in various engineering and managerial capacities. He currently teaches two graduate-level software engineering courses (XML, Distributed programming using Jini networking technology) in Brandeis University's continuing education program in Massachusetts.

(JV6) Using SWT Under Eclipse
TO BE ANNOUNCED

Wednesday March 19, 2003 11:00 A.M. - 11:50 A.M.

The release of the Eclipse project introduced a new windowing toolkit, the Standard Widget Toolkit. Discover what all the fuss is about with the Standard Widget Toolkit and why everyone is talking about it. This session will detail the underlying windowing infrastructure of the SWT Toolkit.

Register by Phone 201 802-3058

ment for the flagship Constellar Hub product. Nigel spent over eight years at Oracle, architecting and delivering Oracle's Accounting products and then moving on to worldwide performance consulting and CASE development assignments.

(JV10) Working with Data the JDO Way

PATRICK LINSKEY, SOLARMETRIC

Thursday March 20, 2003 11:00 A.M. - 11:50 A.M.

Java Data Objects is an alternative way of looking at your data compared to JDBC. Looking at your data in an object-oriented way offers many advantages over the sequential manner of JDBC. Learn how JDO can be adopted for your own requirements through practical examples.

BIO: Patrick Linskey manages and drives SolarMetric's technology development as vice president of Engineering. Patrick has been intimately working with JDO for nearly two years and has been involved in object/relational mapping for over four years. Over the past year, Patrick has evangelized the JDO specification at local Java User Groups and software symposiums throughout the world.

(JV11) Enterprise: The Next Generation

MARK HAPNER, SUN MICROSYSTEMS

Thursday March 20, 2003 3:00 P.M. - 3:50 P.M.

With the latest release of the J2EE 1.4 framework, a whole new suite of APIs have been added to this already comprehensive edition. Chances are, there are goodies lurking in there you didn't even know about. This session will take a look at unearthing some of the gems of the J2EE framework.

BIO: Mark Hapner is a Sun Distinguished Engineer and is currently lead architect for Java™ 2 Platform, Enterprise Edition (J2EE™). He has guided the overall architecture for J2EE 1.2, 1.3 and now the upcoming 1.4 release. In March of 1996, he joined Sun's JavaSoftware Division to participate in the development of the Java database connectivity API (JDBC). Following that he was co spec lead of the Enterprise JavaBeans specification and spec lead of the Java Message Service specification.

Prior to his work on Java enterprise APIs, Mark was a member of Sun's Object Services Group were he wrote several of Sun's initial CORBA object services submissions and worked on the integration of object oriented and relational databases with Sun's ORB.

(JV12) Overcoming the Challenges of J2ME

DR. JEFF CAPONE, ALIGO

Thursday March 20, 2003 4:00 P.M. - 4:50 P.M.

J2ME has brought the power of Java to the mobile space. However, it is fair to note that not all the JDK is available to you. Discover what is and what isn't available to you and how you can potentially work around some of what seems to be "must have" tools.

BIO: Dr. Jeff Capone has devoted his career to researching wireless and wireline networks and applications. As Aligo's CTO, Jeff leads the technology development and is the principal architect of the innovative M-1 Server. Prior to leading Aligo's engineering team, he was an assistant professor at Arizona State University and director of the Network Engineering and Wireless Telecom Lab.

(JV7) Unlocking the Power of XML

HITESH SETH, ikigo

Wednesday March 19, 2003 3:00 P.M. - 3:50 P.M.

There is more to XML than just one block of String. Understand the difference between a DTD and a schema and the APIs you can utilize within Java that will bring the power of XML to your Java development.

BIO: Hitesh Seth is the chief technology officer of ikigo, Inc., a provider of XML-based Web services monitoring and management software. A freelance author and well-known speaker, he regularly writes for technology publications on VoiceXML, Web services, J2EE and Microsoft .NET, wireless computing, and enterprise/B2B integration. He is also the editor-in-chief of *XML-Journal.*

(JV8) Integrating Java and .NET

DEREK FERGUSON, EXPAND BEYOND CORPORATION

Wednesday March 19, 2003 4:00 p.m. - 4:50 p.m.

Two technologies that aren't often mentioned in the same breath are Java and Microsoft. However, it is the rare developer who is able to completely avoid either of these two worlds nowadays. In this presentation, we will examine several tools both free and commercial that can be used to bring these two development platforms together. Some knowledge of Java and Windows development is advisable.

BIO: Derek Ferguson is chief technology evangelist for Expand Beyond Corporation (www.xb.com), the worldwide leader in mobile software for enterprise management. He is also editor-in-chief of *.NET Developer's Journal* and author of the book *Mobile .NET.*

(JV9) Writing SOAP Services

NIGEL THOMAS, SPIRITSOFT, INC.

Thursday March 20, 2003 9:00 A.M. - 9:50 A.M.

J2EE 1.4 mandates the use of JMX to manage compliant products. Early adopters of JMX mostly use protocols like RMI and HTTP to communicate between the management application and the agents it is managing. This presentation describes how JMS is utilized to manage large deployments of managed agents in the Enterprise and optimize the delivery of alerts and notifications in complex environments. The processing of management alerts, using open source components (such as Jelly) to provide an Event-Condition Action (ECA) framework, will also be discussed.

BIO: Nigel Thomas joined SpiritSoft as director of product marketing in April 2001. Prior to SpiritSoft, Nigel spent five years with EAI pioneer Constellar, serving in consulting, support, sales support, and development roles. He became product architect and then director of product manage-

Web Services Track

The Web Services track focuses on issues and topics that are at the forefront of development efforts in Web services. Although the current specifications provide a minimum set of protocols, issues such as security, transaction management, service management and coordination remain in flux. This track presents some of the leading authorities in the field on these urgent topics and addresses all of the questions that currently concern designers, developers and consumers of Web services

(WS1) Web Services Infrastructure

CARL SJOGREEN, BEA

Tuesday March 18, 2003 9:00 A.M. - 9:50 A.M.

Web services have evolved from an over-hyped vision of an interconnected world to a set of real standards and technologies that can solve real-world problems in the enterprise. Building, deploying, and managing Web services in an enterprise-class environment, however, still raises many questions about the infrastructure on which those Web services are deployed.

This session will introduce several real-world Web services case studies, the requirements placed on Web services infrastructure, and several hands-on examples of enterprise-class Web services implementations that address the issues raised above.

BIO: Carl Sjogreen is product manager for BEA WebLogic Workshop, BEA's latest Java innovation and an integrated development environment for building Web services. He has been involved with XML, Web services, and developer tools since 1998, when he founded Transformis, a software startup specializing in XML tools. Passionate about the power of XML and bringing new technologies to the masses, he is a key contributor to growing the BEA WebLogic development community.

(WS2) Web Services Management

JAMES PHILLIPS, ACTIONAL

Tuesday March 18, 2003 11:00 A.M. - 11:50 A.M.

Service-oriented architectures (SOAs) have been held out for years as a substantially more cost-effective and flexible approach to architecting enterprise software systems than historical strategies such as monolithic system design or tightly coupled client/server approaches. Many customers and industry observers believe Web services technology finally makes possible the widespread adoption of the SOA approach. But while Web services and SOAs substantially ease the application development and integration burden, they bring with them a new collection of management challenges. In this session, you will learn about the essential management criteria for growing and sustaining a "mission-critical" service-oriented architecture.

BIO: In his role at Actional, James has worldwide responsibility for Actional's product and market strategy and market execution. Prior to joining Actional, he served as CSO and vice president of product marketing and business development with Ensim Corporation. James is a frequent speaker and editorial contributor on Web services–related issues and serves on the international advisory board of *Web Services Journal.*

(WS3) Strategies for Using Databases in a World of Web Services

MIKE LEHMANN, ORACLE

Tuesday March 18, 2003 3:00 P.M. - 3:50 P.M.

Databases continue to be at the core of most IT infrastructures yet the knowledge of how they play in the world of SOAP-based Web services is less well understood. This presentation covers strategies for publishing database components as Web services and, as important, techniques for consuming Web services within databases. Find out how to plug your database infrastructure into the world of Web services.

BIO: Mike Lehmann has worked in the IT industry for over 12 years as a developer, consultant, and project manager in emerging technologies. His current areas of focus include J2EE and Web services. Mike frequently authors articles and papers on Web services and speaks regularly at industry events.

(WS4) Using Web Services to Integrate J2EE and .NET Enterprise Applications

**ODYSSEAS PENTAKALOS, PH.D.,
SYSNET INTERNATIONAL, INC.**

Tuesday March 18, 2003 4:00 P.M. - 4:50 P.M.

The Web services phenomenon promises to resolve all interoperability issues through an open architecture that is based on widely accepted industry standards such as SOAP, WSDL, and HTTP. This session will explore component reuse across the two major enterprise application platforms, J2EE and .NET, using Web services as the medium of interaction. Through examples, this tutorial will cover the current state of interoperability between J2EE and .NET and will focus on best practices and issues that arise. The Axis SOAP implementation will be used in this session as the Web services platform for exposing J2EE components.

BIO: Odysseas Pentakalos is vice president of SYSNET International, Inc., where he focuses on architecture, design, and development of large distributed systems that utilize Java and J2EE technologies. He holds a Ph.D. in computer science, is the author of the *Windows 2000 Performance Guide*, has published over two dozen papers in conference proceedings and journals, and is a frequent speaker at industry conferences.

(WS5) Combining BPM and BRM Technologies: A Major Step Towards Corporate Agility

HENRY BOWERS, ILOG

Wednesday March 19, 2003 9:00 A.M. - 9:50 A.M.

Corporate agility: the ability to quickly respond to unexpected change inside and outside the corporation, is a coveted quality of the modern enterprise. This presentation discusses how corporate agility is driven by technical agility. It introduces business rule management (BRM), business process management (BPM), and Web services technologies, and explains how together these technologies provide an enabling foundation for technical agility.

BIO: Henry Bowers has spent more than 15 years in the high-tech sector, building and managing software products for both private industry and government. He has more than 7 years' experience working with rules-based systems and business rules in general. Henry is currently a product manager for business rules at ILOG.

(WS6) Web Services for Real-Time Data Access in an Industrial Setting

STEPHAN VAN DIJCK, ABB/SKYVA

Wednesday March 19, 2003 11:00 A.M. - 11:50 A.M.

Real-time processes deliver and require data in order to participate in business transactions. We will present an application under development at ABB for linking real-time process control with business processes via Web services, and explore how plant floor control can be a consumer and producer of Web services. We will use solutions from ABB, IBM WebSphere, and Microsoft .NET.

BIO: Stephan Van Dijck holds a master's degree in engineering mechanics from K.U. Leuven, Belgium, and a degree in business management from I.A.G. U.C., Louvain, Belgium. He has worked as a process engineer at the EXXON facility in Antwerp, Belgium, where he supervised logistics planning and optimization of loading and unloading operations with Honeywell Europe as an application and software development manager, product line marketing manager for Batch control systems, and marketing manager for the chemicals and pharmaceuticals business unit Europe. In 1998 he joined SKYVA, a provider of e-manufacturing solutions by making use of Web services.

(WS7) Web Services Architecture: The Next Big Spec, from the Mouths of the W3C Authors

ERIC NEWCOMER, IONA (MODERATOR)

Wednesday March 19, 2003 3:00 P.M. - 3:50 P.M.

The World Wide Web Consortium (W3C) was created to develop interoperable technolo-

gies and to provide an open forum for discussion about the future of the Web. A significant effort currently underway within the W3C's Web Services Activity is the development of a Web Services Architecture Specification to help guide the future direction of Web services.

This panel session will discuss the role of the W3C in the development of Web services standards through this effort. The panel includes cochairs of the working group and editors of the spec, with representatives from BEA Systems, Contivo, IBM, IONA, and Software AG. Providing conference attendees with an opportunity to learn more about what the W3C is doing with regard to Web services, this panel will demonstrate the benefits and examine the challenges of working together to develop an open standard architecture.

BIO: In the role of chief technology officer at IONA, Eric is responsible for IONA's technology roadmap and the direction of IONA's Orbix E2A e-Business Platforms as relates to standards adoption, architecture, and product design. Eric joined IONA in November 1999, and most recently served as IONA's vice president of engineering, Web Services Integration Products. Eric is a member of the XML Protocols and Web Services Architecture working groups at the W3C and IONA's Advisory Committee representative to UDDI.org.

(WS8) – Web Services: The Next Steps After the Hype
CLAIRE DESSAUX, ORACLE
Wednesday March 19, 2003 4:00 P.M. - 4:50 P.M.

Because Web services are still very new, many companies have yet to understand what Web services mean for their line of business and how they can make the most of this evolutionary technology. This session will highlight real-life examples where Web services have been successfully implemented and examines where services make sense, especially as they relate to integration. It will conclude with recommendations on how to best plan for a Web service strategy.

BIO: Claire Dessaux joined Oracle Alliances in 1994 with a main focus on helping partners implementing Oracle technology. Since then she has worked in various capacities on the incorporation of J2EE, XML and Web services into Oracle's product lines.

(WS9) Best Practices for Web Services Development
CHRIS PELTZ, HP
Thursday March 20, 2003 9:00 A.M. - 9:50 A.M.

There has been a fair amount of hype around Web services and the benefits they can bring to an organization. For many, it is difficult to determine how to best get started with Web services. This talk will outline a number of first steps that can be taken to begin planning a Web services pilot. It will also address a series of design guidelines, patterns, and architectural recommendations to enable an organization to build robust, flexible, and secure services. The talk will conclude with a look at tips and techniques for developing, testing, and managing Web services.

BIO: Chris Peltz is a senior software consultant within HP's Developer Resources Organization. He provides technical and architectural consulting to enterprise customers on J2EE, Web services, and mobile development. Chris has over 10 years of software experience in object-oriented technologies, 4GL development, GIS, and Web applications design.

(WS10) Web Services Startups: Telltales of the Future
SIMEON SIMEONOV, POLARIS VENTURE PARTNERS
Thursday March 20, 2003 11:00 A.M. - 11:50 A.M.

You must have seen them; they're everywhere. Web services startups are popping up like mushrooms after rain. Or has the Web services hype created a microcosm of startup activity that is going to mirror the boom and bust cycle of the late '90s? As a Web services technologist, where should you focus your energies? As a customer, how should you choose your vendor? As an investor, where should you put your money to work? In this presentation you'll get an overview of startup activity in the Web services space within the context of emerging industry dynamics and evolving customer needs. We will cover Web services tools, appliances, runtimes, integration, security, testing, and management.

BIO: Simeon Simeonov is a principal at Polaris Venture Partners, a leading early-stage venture firm, where he focuses on investments in information technology. Prior to joining Polaris, Sim was chief architect and vice president of emerging technologies at Macromedia, where he led the development of Web services technologies and platform infrastructure for next-generation Internet applications. Prior to that, he was a founding member and chief architect at Allaire, where he was responsible for the ColdFusion application server and cross-product technology strategy.

(WS11) Open Standards for Web Services Messaging
DAVE CHAPPELL, SONIC SOFTWARE
Thursday March 20, 2003 3:00 P.M. - 3:50 P.M.

Web services hold the promise of driving down the cost and complexity of application integration both between internal systems and between business partners. But unless Web services communications are made reliable, organizations will not be able to trust them for mission-critical operations, such as complex business-to-business transactions or real-time enterprise integration.

In this session, Dave Chappell, a coauthor of the WS-Reliability specification, will examine the emerging Web services reliability standards, including a detailed discussion on WS-Reliability, a new specification for open Web services messaging.

BIO: Dave Chappell, vice president and chief technology evangelist for Sonic Software, has over 18 years of experience in the software industry covering a broad range of roles including R&D, code-slinger, sales, support, and marketing. Dave has a strong passion for shaping the future of technology and enjoys sharing his knowledge and experience with others.

(WS12) – Web Services Security
MARC CHANLIAU, NETEGRITY
Thursday March 20, 2003 4:00 P.M. - 4:50 P.M.

Web services deployments typically use transport-level security for authentication and application-based access control for authorization. This presentation shows the limitations of transport-level security and introduces the XML technologies that complement the transport-level approach to provide a secure authentication solution. This presentation also covers the issues involved in providing access control in back-end applications and suggests a better, centralized approach to abstract authorization information out of back-end applications and into a single point of control using XML-based query technologies.

BIO: Marc Chanliau is a product manager for Netegrity, Inc. He has been in the software industry for over 20 years in many different capacities. Marc started the OASIS Security Services Technical Committee which culminated in the adoption of SAML as an official OASIS standard in November 2002. He holds an MS in linguistics from the University of Paris-Jussieu, France.

.NET Track: .NET Edge 2003 East – International .NET

Microsoft .NET Track

Microsoft .NET represents a major evolution in how applications are developed, deployed, and managed on the Microsoft platform. The .NET Framework gives developers an object-oriented development environment for building all types of applications, including desktop, client/server, dynamic Web page, wireless devices, server-based as well as complete support for XML Web services and the related XML standards. The sessions in the .NET Track will give you a broad as well as deep understanding of the capabilities in the .NET Framework and how applications built on .NET are easily integrated with applications running in heterogeneous environments, including mainframe, UNIX, and J2EE platforms.

(NT1) .NET Framework Overview
BOB FAMILIAR, MICROSOFT
Tuesday March 18, 2003 9:00 A.M. - 9:50 A.M.
This session will introduce the architecture and technologies within the .NET Framework, including the Common Language Runtime, ASP.NET, and ADO.NET, as well as XML and SOAP support. Integration with COM and COM+ services will be covered, as well as a discussion of the .NET Framework SDK.

BIO: Bob Familiar is a Microsoft Architectural Engineer and Certified Microsoft Solution Developer with over 14 years of experience in software engineering. He has developed solutions in C, C++, Java, SQL, and Visual Basic, and has over 7 years of experience using Microsoft Development Tools. Bob holds a patent for technology that maps relational database tables to object-oriented software components. His current areas of interest include object oriented analysis and design of distributed object models and distributed computing using COM+.

(NT2) Introduction to ASP.NET
RUSS FUSTINO, MICROSOFT
Tuesday March 18, 2003 11:00 A.M. - 11:50 A.M.
We will look at the next version of Active Server Pages, ASP.NET, a major upgrade to the current version of ASP. Xcopy deployment; control-based encapsulation; clean separation of code from HTML; strongly typed, compiled languages; and event-based page processing will change the way you develop UI for server-based Web applications. This session covers the ASP.NET Page Framework from the ground up: architecture; ASP.NET syntax; server controls; control families: intrinsic controls, list controls, and rich controls. Finally, this session will cover business objects, Web services, and migration issues.

BIO: Russ Fustino is a Microsoft Principal Technology Specialist and a Microsoft Certified Professional with over 20 years of software development experience. He has an expertise in developing Visual Basic and Web-based solutions using Microsoft tools. Prior to Microsoft, Russ was a VB instructor, headed up a

development department and created several client/server application and system software products. Russ's specialties include development with VB.NET, XML Web services, ASP.NET, and debugging.

(NT3) Introduction to VB.NET
RUSS FUSTINO, MICROSOFT
Tuesday March 18, 2003 3:00 P.M. - 3:50 P.M.
Looking to learn the latest release of Visual Basic, VB.NET? Wait no longer and jump right in! See first hand why VB is the language of choice for developers. Whether you are changing from another language or just re-tooling, make sure you hit this session. You won't regret it! We will cover the Visual Studio Integrated Development Environment, Win Forms, Web forms and ASP.NET, classes and objects, XML Web services, ADO.NET introduction for data access and debugging.

BIO: Russ Fustino is a Microsoft Principal Technology Specialist and a Microsoft Certified Professional with over 20 years of software development experience. He has an expertise in developing Visual Basic and Web-based solutions using Microsoft tools. Prior to Microsoft, Russ was a VB instructor, headed up a development department, and created several client/server application and system software products. Russ's specialties include development with VB.NET, XML Web services, ASP.NET, and debugging.

(NT4) How to Develop an End-to-End .NET-Connected Application
ALLAN DE COSTA PINTO, MICROSOFT
Tuesday March 18, 2003 4:00 P.M. - 4:50 P.M.
This session will show you how to design and create a sample application using multiple Microsoft .NET Enterprise Server products. We will show you how to integrate .NET technologies into an effective solution. The following products and technologies will be covered in this session: VS.NET, ASP.NET, VB.NET, C#, XML Web services, BizTalk, and SQL Server 2000. (Note: The demos in this session have been updated for VS.NET RTM version.)

BIO: Allan da Costa Pinto is a Microsoft Certified Application Developer for .NET and works for Microsoft Consulting in the Connecticut Area. Allan focuses on architecting and building solutions using Microsoft Visual Studio® .NET and XML Web services.

(NT5) .NET, the Virtualized Execution Engine
YAHYA MIRZ, AURORA BOREALIS
Wednesday March 19, 2003 9:00 A.M. - 9:50 A.M.
With .NET, Microsoft has introduced a major evolution in their computing platform. At the core of .NET, is a language agnostic runtime, currently being standardized by ECMA. This effort is called the Common Language Infrastructure or the CLI. The objective of this presentation will be to provide insight into the design decisions that led to the CLI's Execution Engine and their rationale. Additionally, I will explain through an example, how a component works it's way through the Virtualized Execution Engine of the CLI.

BIO: Yahya Mirza is striving to balance commercial development with component-based language research. Since 1999, Mr. Mirza has been working on .NET projects at Microsoft, Source Dynamics and Safeco Life. His passion lies in language design for music and computer animation.

(NT6) Introduction to DotGNU
BARRY FITZGERALD, DOTGNU
Wednesday March 19, 2003 11:00 A.M. - 11:50 A.M.
This session will feature a brief overview of the many GNU projects currently implementing and extending ECMA CLI standards. These projects include SEE, phpGroupWare, Portable.NET, Web Services, and Virtual Identities.

BIO: Barry Fitzgerald is one of the cofounders of the DotGNU Free Software Project to create a 100% open source GNU implementation of the ECMA Common Language Infrastructure (CLI). He is also a member of the Steering Committee for DotGNU.

(NT7) Intro to SSCLI
YAHYA MIRZ, AURORA BOREALIS
Wednesday March 19, 2003 3:00 P.M. - 3:50 P.M.
In early 2002, Microsoft released a "Shared Source" implementation of the .NET

Register by Phone 201 802-3058

Common Language Infrastructure available on BSD UNIX, Linux, Mac OSX, and the Windows platform codenamed "Rotor". For enthusiasts, Rotor provides an opportunity to understand the .NET technology at a deeper level. For language designers, Rotor can serve as an effective runtime core for experimentation at the language feature level. For compiler and virtual machine researchers, Rotor provides a context for applied research into alternative object representations, method dispatch, garbage collectors, JIT compilers, etc. My goal will be to provide an introduction into the Rotor code base.

BIO: Yahya Mirza is striving to balance commercial development with component-based language research. Since 1999, He has been working on .NET projects at Microsoft, Source Dynamics and Safeco Life. His passion lies in language design for music and computer animation.

(NT8) Mobile Development with the Compact Framework

BRAD MCCABE, INFRAGISTICS

Wednesday March 19, 2003 4:00 P.M. - 4:50 P.M.

With the release of the Compact Framework, Microsoft has brought mobile development to the masses. In this session we will look at some of the differences and similarities between the .NET Framework and the Compact Framework. We will also examine the consideration that must be factored into the architecture and development of mobile application such as dealing with user interface considerations and effective mobile data strategies

BIO: Prior to joining Infragistics, Brad McCabe served as systems architect for the network solutions development team at Verizon Communications. Concurrently, Brad held the position of lead .NET Evangelist within Ajilon Consulting and was responsible for content and delivery of material for the national Inside .NET tour. Brad has been working as a senior developer and a software engineer, and his current responsibilities include developing reference applications and working with enterprise customers on project implementation.

(NT9) Best Practices for .NET Development

JOE STAGNER, MICROSOFT

Thursday March 20, 2003 9:00 A.M. - 9:50 A.M.

This advanced session covers the wide array of best practices for the most productive .NET development topics. We'll start with an overview of design guidelines for .NET, ensuring that developers write consistent, predictable code that integrates well between languages. We will cover memory management and the garbage collector as well as the best methods for cleaning up unmanaged resources. A section will cover how to best use ADO.NET to access relational data, and how to assimilate that data with the .NET XML classes. We'll look at tips for creating both components and controls, how to best use the System.NET classes to access HTTP responses, the .NET threading model, and key security concepts will also be examined.

BIO: Joe Stagner is a technical evangelist of Developer Technologies at Microsoft, and has been developing software for 25 years, on Microsoft platforms since DOS 1.0. Joe currently focuses on helping Microsoft's Independent Software Vendors make the most of .NET Developer technologies and also works with medium business developers and the academic community in New England.

(NT10) Best Practices for ADO.NET Development

THOM ROBBINS, MICROSOFT

Thursday March 20, 2003 11:00 A.M. - 11:50 A.M.

One of the biggest changes in data access recently is that common clients and servers on the Internet work in a disconnected, stateless fashion. ADO.NET provides an optimized, XML-based framework for data access in a disconnected environment. With the advent of XML, programming for data access and manipulation, the mechanism is changing from a relational to a hierarchical form. ADO.NET classes read and write XML at their core. The ADO.NET framework fully supports disconnected reads and writes. and we'll discuss the performance implications of this as well as how to use different types of objects in their various states. ADO.NET also has classes that provide optimized performance with SQL Server or other OLE DB data sources. This session covers the best practices on how to handle data access for best performance, error handling, transaction support, and security. We'll discuss the best practices for retrieving, single item, single row and multiple rows of data as well as transaction handling, and look at examples of the performance differences in using DataReaders versus DataSets This session will also cover the best ways to cache data from data sources and the transformation to XML.

BIO: Thom Robbins is a senior technology specialist with Microsoft. He is a frequent contributor to various magazines include *.NET*, *Visual Studio.NET* and the *Web Services Journal.* Thom is also a frequent speaker at a variety of events that include VS Live and others. When not writing code and helping customers, he spends his time with his wife at their home in New Hampshire.

(NT11) How to Debug with .NET

TOBY DENBOW, STAR IT

Thursday March 20, 2003 3:00 P.M. - 3:50 P.M.

This session shows how the rules for debugging have completely changed with the advent of Microsoft .NET Framework. It will cover the core debugging concepts using the Microsoft .NET Framework. In addition, this session will discuss many advanced features that allow you to write better diagnostic and error-handling code.

BIO: Toby Denbow is VP of technology at STAR Information Technology. He has been a featured speaker at several Microsoft and industry trade shows. Toby has been working with .NET for over a year and was trained directly by the .NET developers in Redmond. He works with a wide variety of customers in various emerging technologies and has personally trained over 500 developers on Visual Studio.NET.

(NT12) XML and Web-Enabling Legacy Applications Using BizTalk

MIKE CRAMER, SENIOR PRINCIPAL TECHNOLOGY SPECIALIST, MICROSOFT

Thursday March 20, 2003 4:00 P.M. - 4:50 P.M.

This session examines how to integrate legacy line-of-business applications using BizTalk 2000 Server. It will examine BizTalk Server tools that allow you to create the components necessary to interface to legacy systems. This includes a scenario with four phases that demonstrates integration by using industry-standard XML and EDI file formats, delimited and positional files, and COM bridges.

BIO: As a member of the New England BSG Team and focusing on integrating heterogeneous applications, Mike Cramer works with existing and prospective Microsoft customers in New England on adopting the .NET platform. Mike joined Microsoft during the 1995 acquisition of Netwise Inc. (Boulder, CO). Prior to the acquisition, Mike worked for Netwise for approximately three years as a consultant and later consulting manager.

XML Technology Track

Whether you're looking to understand different XML standards, application techniques, or development tools; or using XML to develop the next generation of Web applications and services, the XML Track is your ultimate training, collaboration, and innovation ground. Sessions include fast-track, in-depth training on XML Schemas and XSL-FO. We will update you on standards development and offer a comprehensive review of the various technologies related to XML that are essential for today's IT manager. The XML Track is armed with real-world applications of XML in financial services, life sciences, enterprise and B2B integration, and mobile computing. We will discuss new developments around XForms, a recent W3C Standard which marks another era of standards-based application development; XPath and XSLT 2.0 XML; and Query.

The XML Track explores the technology and standards, real-world applications, and trends which will set the course for the future.

(XM1) XML: A Manager's Guide

JP MORGENTHAL, SOFTWARE AG

Tuesday March 18, 2003 9:00 A.M. - 9:50 A.M.

This session will provide an essential introduction to XML from a manager's perspective. As more and more IT projects utilize XML and its derivatives as fundamental technologies, it is key for today's manager to be aware of the various ingredients of XML. From core XML processing, transformation, metadata definition and schemas, applications in Web, wireless and speech applications, Web services, industry-standard vocabularies, and more, the session will offer a comprehensive review of the various technologies related to XML that today's IT Manager must be aware of.

BIO: JP Morgenthal serves as the chief services architect for Software AG, Inc. He is an internationally prominent authority on XML with more than 15 years of experience designing, developing, and analyzing software and technology. In his role as chief services architect, JP will explore and manage the design of complete professional services solutions based on Software AG technology and partner products in existing and emerging industries.

(XM2) OASIS Standards Update

KARL F. BEST, OASIS

Tuesday March 18, 2003 11:00 A.M. - 11:50 A.M.

OASIS is a not-for-profit, global consortium that drives the development, convergence, and adoption of e-business standards. Members of OASIS are providers, users, and specialists in standards-based technologies, and include organizations, individuals, and industry groups. The OASIS standards process can best be described as open, lightweight, and independent.

This session will address, OASIS Initiatives, current TC status, OASIS Technical Agenda, collaborative work at OASIS, EbXML, ISO/IEC/ITU/ECE Memorandum of Understanding (MoU) for electronic business, standards, liaison memberships in various ISO TCs and ISO/IEC JTC1 SCs and more ongoing, sponsor of Interoperability Summit, vendors submit specs to OASIS, dot-orgs move to OASIS, convergence at OASIS, and current Technical Committees,

BIO: Karl F. Best is the director of technical operations for OASIS, where he is responsible for managing the consortium's industry standards efforts. He is a former chief strategy officer and board member of the consortium, and has been involved in the XML/SGML industry since the early 1990s. Karl has been a member of a number of industry standards committees for SGML, XML, and associated standards, and a speaker at many industry conferences. He was previously employed by Novell, Adobe, and Sun Microsystems, designing and implementing electronic documentation systems.

(XM3) A Definitive Introduction to XML Schemas

HITESH SETH, IKIGO

Tuesday March 18, 2003 3:00 P.M. - 3:50 P.M.

XML Schemas have emerged as the standard way of representing XML vocabularies. Schemas extend the basic mechanism for defining the structure and content of XML documents established by Document Type Definitions (DTDs) to include rich data-type information and features such as simple/complex types, extensions/restrictions, etc. Get started with XML Schema development with this introduction to the standard and best practices for XML Schema design. We will also preview how XML Schema compares to RELAX NG.

BIO: Hitesh Seth is the chief technology officer of ikigo, Inc., a provider of XML-based Web services monitoring and management software. A freelance author and well-known speaker, he regularly writes for technology publications on VoiceXML, Web services, J2EE and Microsoft .NET, wireless computing, and enterprise/B2B integration.

(XM4) XML in Print: XSL-FO

FRANK NEUGEBAUER, IBM

Tuesday March 18, 2003 4:00 P.M. - 4:50 P.M.

Today developers frequently use XSLT to convert XML into markup languages such as HTML. XSL Formatting Objects (XSL-FO) promises to abstract the details of rendering and allow developers to specify the layout and semantic properties of presentation regardless of rendering. This presentation will provide a thorough introduction to XSL-FO and how it relates to the other XML processing standards and will highlight the benefits of XSL-FO in XML application development. Using demonstrations, the session will highlight the features that XSL-FO provides for creating an open standards–based print process and its applications.

BIO: Frank Neugebauer is a consultant in the Insurance Solutions division of IBM Global Services, specializing in distributed Java solutions based on IBM's Insurance Applications Architecture (IAA). He is also a freelance writer and has contributed several articles to leading industry publications.

(XM5) XML Security Integration Challenges

PHIL STEITZ, AMERICAN EXPRESS

Wednesday March 19, 2003 9:00 A.M. - 9:50 A.M.

This session will highlight the key security integration challenges faced by application teams developing enterprise solutions using loosely coupled, XML-based interfaces and service-oriented architectures. We will discuss how emerging standards and technologies address the practical security problems faced by development teams, the gaps that still remain, and the tradeoffs and compromises that architects and developers need to make to implement secure solutions today.

BIO: Phil Steitz is vice president, e-commerce applications development, at American Express. He has over 20 years of experience as a developer, architect, and technology leader involved in distributed systems development. Before joining American Express, Phil served as a middleware architecture consultant, designing large-scale distributed systems for enterprise customers. He holds a PhD in mathematics from the University of Maryland.

(XM6) Case Study: XML in the Life Sciences

TIM MATTHEWS, IPEDO

Wednesday March 19, 2003 11:00 A.M. - 11:50 A.M.

The importance of XML-based information in the health care and biotech industries has grown tremendously over the last two years, from R&D to clinical trials to manufacturing. Life sciences companies today are required to transfer and share huge quantities of information among the myriad of researchers and partners involved in the product development life cycle. The future of the life sciences market will be influenced by how well companies acquire, share, and apply knowledge to exploit the wealth of new opportunities while minimizing the deluge of new risks and costs. This session will provide a case study of how one life sciences company is implementing XML for a competitive advantage. It will also discuss how XML Management technology was used to provide scientists and analytic applications with the ability to efficiently search and analyze the data using sophisticated queries.

BIO: Tim Matthews has extensive experience in high-tech engineering, marketing, and sales. Prior to cofounding Ipedo in 2000, he was director of product marketing at RSA Security, where he oversaw a line of developer security products and a line of security infrastructure servers. Previously, Tim worked in international sales and business development at Digital Equipment Corporation in Tokyo and Irvine, California.

(XM7) Using XML for EAI: Best Practices

DAN ENACHE, TIBCO SOFTWARE

Wednesday March 19, 2003 3:00 P.M. - 3:50 P.M.

According to Gartner Group, EAI is the hottest IT market. Even in a slow economy, the EAI market is expected to grow at a steady pace, faster than most of the other sectors. How can you take advantage of EAI and help your company reap the benefits? What are the best practices in using XML to implement large EAI systems? What are the pros and cons in using XML as a "lingua franca"? How are people in the trenches using XML and EAI to build systems that work?

BIO: Dan Enache is a senior software architect for TIBCO Software. He is an expert in large integrations both in the EAI space and on the Web, as well as large portals and Single Sign-On integration solutions. Dan has extensive experience with high volume/high availability transactional/financial systems, and a great deal of expertise in implementing large security systems. He is a J2EE developer and Sun Java Architect

(XM8) Delivering Web Services to Mobile Clients with XML Transformation Frameworks

PAUL LIPTON, COMPUTER ASSOCIATES

Wednesday March 19, 2003 4:00 P.M. - 4:50 P.M.

Much of the discussion on Web services focuses on the server side, but careful consideration of the delivery of Web services to client platforms is also necessary. Important technology and business trends will increase the diversity of client platforms, and will require that you consider mobile devices at the earliest stages of design. In fact, proper visualization, aggregation, and delivery of Web services in the new world of highly variable client platforms will be essential for the ultimate success of many projects.

BIO: Paul Lipton is the Web services technology leader for the field services organization and a technology strategist in the Office of the CTO. He has been an architect and developer of enterprise systems for more than 20 years, and has worked closely with key CA customers to architect distributed solutions using J2EE, .NET, wireless, and Web services technology.

(XM9) XML, Ontologies & the Semantic Web

AYESHA MALIK, OBJECT MACHINES

Thursday March 20, 2003 9:00 A.M. - 9:50 A.M.

The Semantic Web has generated much talk ever since Tim Berners-Lee, the inventor of the World Wide Web, first mentioned it a few years ago. Get the architecture details behind a Semantic Web. In particular, the session links the study of ontologies for modeling knowledge representation with the requirements of a Semantic Web. Track W3C's standardization activity in building XML standards for the Semantic Web including and RDF and OWL (Web Ontology Language).

BIO: Ayesha Malik is a senior consultant for Object Machines, a software engineering firm providing Java technology and XML solutions to businesses. Ayesha has worked extensively on large XML and messaging systems for companies such as Deutsche Bank and American International Group (AIG). Most recently, she has been researching new ways to make schemas extensible and object oriented.

(XM10) XQuery

MIKE CHAMPION, RESEARCH AND DEVELOPMENT SPECIALIST, SOFTWARE AG

Thursday March 20, 2003 11:00 A.M. - 11:50 A.M.

As the ubiquitous data description and encapsulation standard, XML has quickly evolved into a container for all kinds of data. It is imperative that a standardized and flexible query and processing language be available to utilize the immense wealth of data that XML stores represent. Currently, even though established standards such as XSLT/XPath are available for transforming XML documents, the development around XQuery language has demonstrated the need for a flexible language for extracting data from XML documents. This session provides an introduction to XQuery language and illustrates its application through real-world scenarios.

BIO: Michael Champion is a research and development specialist at Software AG, working out of Ann Arbor, Michigan. He has been a software developer in the USA for 20 years, working primarily in the area of middleware for client/server document and image management systems. He has been active in the W3C's Document Object Model (DOM) Working Group for more than three years and was an editor of the core XML portion of the DOM Level 1 Recommendation. He is now cochair of the Web Services Architecture Working Group.

(XM11) XPath & XSLT 2.0

KURT CAGLE, CAGLE COMMUNICATIONS

Thursday March 20, 2003 3:00 P.M. - 3:50 P.M.

XPath and XSLT 1.0 are about to get a face-lift. Most of the improvements to the languages center on support for XML Schema, XML's official type system, although there are other compelling improvements that make the languages more usable as well. We'll walk you through "what's new" in both XPath and XSLT 2.0 and provide several examples using a reference implementation.

BIO: Kurt Cagle is the president of Cagle Communications, a consulting firm specializing in XML-related technologies, and is the author of 14 books on XML, XSLT, XQuery, and SVG. He is a columnist for *XML Magazine*, and publishes an e-newsletter that can be subscribed to from The Metaphorical Web.

(XM12) Third-Generation XML Tools

MICHAEL LEVENTHAL

Thursday March 20, 2003 4:00 P.M. - 4:50 P.M.

XML has become the ubiquitous infrastructure for Web services and a host of other software applications. The tools for building XML applications have progressed from the first generation of "hackerware" to a mature second generation of public and private domain tools providing SAX and DOM programming interfaces, robust parsing, well-formed and validation checking, and transformation with XSLT. Is there anything more to come? This presentation looks at two areas where interesting, third-generation tools are beginning to emerge: XML programming objects and streaming. With XML programming objects developers deal directly with document-specific classes derived from XML Schemas rather than generic document structures, as in the DOM or document events in SAX. JAXB reference implementation and Castor are two tools in this area, but there are also interesting developments coming that tightly mesh XML objects into programming languages. Streaming is based on the event model of XML parsing but goes beyond SAX with sophisticated techniques for analysis and processing of XML documents with ultra-low memory usage and high throughput. Streaming is being used in SOAP processors, search engines, entity resolvers, and transformation tools. Examples include the STX and XST transformation engines, Apache AXIS, and sequential XPATH implementations. At the end of this session we'll discuss ideas for next-generation XML tools

BIO: Michael Leventhal led the team that architected and developed a document-style SOAP framework for Commerce One. He has led numerous projects in the area of Web applications and infrastructure and XML (and SGML) over the last 10 years and wrote the first book on XML software development for the Internet in 1998.

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

(VN2) The XMLSPY 5 Enterprise Edition Development Environment

TRACE GALLOWAY, CORPORATE SALES MANAGER, ALTOVA

Tuesday March 18, 2003 11:00 A.M. - 11:50 A.M.

XML-related technologies have begun to reach critical mass in many areas of business today. From e-commerce solutions to data integration initiatives to content authoring and publishing, XML related technologies are being used to meet many of the present and future business challenges. Altova™'s XMLSPY 5 Enterprise Edition Development Environment is ideally suited to meet the needs of developers, Web designers, and line of business users that are required to develop and work with XML related technologies. XMLSPY 5 Enterprise Edition is the industry-standard XML Development Environment for designing, editing, and debugging enterprise-class applications involving XML, XML Schema, XSL/XSLT, SOAP, WSDL and Web Services technologies. It is the ultimate productivity enhancer for J2EE, .NET and database developers. In this presentation, attendees will be introduced to many of the features available in the Development Environment.

BIO: Trace Galloway is the corporate sales manager at Altova, Inc, creators of XMLSPY, the award-winning and industry-leading tool suite for XML. Prior to joining Altova, Trace served as the chief evangelist for Infoteria Corporation, a global technology company specializing in the development of B2B-centric, XML-based servers and components. He has presented at numerous industry conferences including Web Services Edge New York, XML Conference Baltimore, and ASP.NET & XML Web Services Solutions Conference. He was co-author of a chapter entitled "Lead Tracking by Web and Email" in the *XML Handbook*" Third Edition.

(VN3) SOAP and Java: Marrying Them Off

ALEXANDER MARLER, PARASOFT

Tuesday March 18, 2003 3:00 P.M. - 3:50 P.M.

 SOAP is a lightweight, XML-based protocol for exchanging structured and typed information between peers in a decentralized, distributed environment. Although SOAP strives to be agnostic with respect to programming languages, some languages facilitate working with SOAP better than others. The combination of Java's strong typing system and its reflection API make Java especially conducive to implementing SOAP-based Web services. The presentation will explore how Java facilitates the automation of activities such as WSDL (Web Services Description Language) generation and SOAP deployment. It will also discuss how these features are influencing the development of Java-based SOAP tools.

BIO: Alexander Marler, software technical consultant, joined Parasoft to provide technical pre and post sales support along with business development for specific product lines. He has over 15 years of sales and technical expertise in the high tech industry. He has been responsible for product and business development at Sybase Inc., Charles Schwab, and Hunt-Wesson Foods. Marler received his Bachelor of Science in Management Information Systems from Washington State University.

(VN5) Process-Centric Enterprises: The Coming Revolution in Web Services–Driven Business Analytics

ERIC PULIER, CHAIRMAN AND CEO, DIGITAL EVOLUTION

Wednesday March 19, 2003 9:00 A.M. - 9:50 A.M.

As major companies transition to Web services and the Service Oriented Architecture (SOA), they are beginning to tap into the SOA's ability to provide greater visibility into business operations in real time. This presentation will explore how exposing the IT functions of a given business process as Web services makes it possible to monitor the activities of that process, even if the process occurs across multiple lines of business and IT systems. The result is what Mr. Pulier terms the "Process-centric" enterprise, which benefits from the functionality of integrated applications and systems without the rigid, time-consuming, and costly process of actual application integration. This session will address these issues and others as he lays out his vision of where enterprise computing is heading.

BIO: Recently named one of 30 e-Visionaries by VAR Business, Eric Pulier is a popular speaker at many elite technology conferences. As CEO and founder of Digital Evolution, he drives the operations and strategic vision of a company that is making the service-oriented architecture revolution a reality. Pulier has been a pioneer in the interactive industry for over 15 years. In 1997, the Presidential Inaugural Committee selected Mr. Pulier to create and execute the Presidential Technology Exhibition in Washington, D.C. He is a graduate of Harvard University.

(VN6) Pattern-Driven Application Development

TOM SHORE, SENIOR PRODUCT CONSULTANT, COMPUWARE

Wednesday March 19, 2003 11:00 A.M. - 11:50 A.M.

OMG's model-driven architecure (MDA) allows organizations to build platform and language neutral models and specifications of functionality. This approach will improve the stability, durability, and reuse of models, as they are not bound to any specific technology. Furthermore, it will boost developer productivity by increasing the code generation capabilities of development tools. As models are translated from a platform independent model (PIM) to a platform specific model (PSM), and further to an implementation (code) model, intelligent code generation engines are able to produce more and better code. Compuware's OptimalJ J2EE development tool is the only implementation of the OMG MDA so far. This presentation discusses how OptimalJ implements OMG's MDA.

BIO: Thomas S. Shore is a senior product consultant with Compuware Corporation's Application Development & Portal Solutions Software Division. Thomas joined Compuware in 1993 and held various positions before focusing on J2EE and related technologies. He has also held various software engineering and consulting positions in the manufacturing, oil & gas exploration and database software markets.

(VN7) Managing the Developer Relationship

MIKE BELLISSIMO, SUN MICROSYSTEMS

Wednesday March 19, 2003 3:00 P.M. - 3:50 P.M.

More and more vendors are developing extensive programs to capture developer interest and loyalty. These programs must adopt a philosophy of managing and enhancing the individual developer's entire experience through the life cycle of the relationship. This includes managing the developer's experience in all places where it touches the vendor. Mike Bellissimo, senior director of Sun Microsystems Software Developer Marketing and Management, will discuss how he believes developers can and should be supported with programs that help them learn, plan, evaluate, and develop their products and services.

BIO: In over a decade at Sun, Mike Bellissimo has managed sales operations for iPlanet and JavaSoft, software training and services for SunSoft, and developer programs and strategy in Sun's market development organization. Previously he managed product training for field and reseller organizations.

(VN10) Model Driven Development of Web Services in UML for the J2ME Platform

BILL GRAHAM, RATIONAL SOFTWARE

Thursday March 20, 2003 11:00 A.M. - 11:50 A.M.

Mobile and wireless clients are an integral part of the vision for Web services. The hardware computing and memory limitations of handheld and wireless devices require the use of lean-and-mean utilities to make Web services practical. This presentation looks at some of the possible solutions in the context of a typical J2ME platform. Models of possible architectures for J2ME applications are explored through the use of UML. Using models to generate code and therefore applications through Model Driven Development (MDD) are also introduced.

BIO: Bill Graham is the embedded Java evangelist at Rational Software. He has over 14 years of experience in working with real-time and embedded systems. He has presented at JavaOne, the Rational Users Conference, and the IEEE International Performance, Computing, and Communications Conference (IPCCC). Bill has a B. Eng. and a M. Eng. from Carleton University, Ottawa.

(VN11) Why Web Services Management?

JON ATKINS, HP

Thursday March 20, 2003 3:00 P.M. - 3:50 P.M.

The popularity of Web services continues to grow because they reduce integration costs and enable greater flexibility and ease in exposing applications as new sources of revenue. However, this benefit is achieved with less secure, less reliable communications. To get the most out of Web services, you must be able to manage and control the new risks that come with them. This lively and informative presentation will answer these questions and reveal the importance of developing Web services with manageability in mind.

BIO: Jon Atkins is the product manager for HP's Web Service Management Platform. He brings 10 years of marketing, sales and advertising experience, primarily in high technology.

Java University℠ Program: Aggressive, code-level training courses for experienced developers using Java™ technology, brought to you by Sun Microsystems, Inc. Attend seminars designed by industry luminaries and recognized experts. Sessions cover Sun certification and Web services technology. Whether you're a beginner or a veteran developer, architect, or software engineer, you'll benefit from these value-packed full-day courses. Register now. Seating is limited.

The Java University℠ program complements this year's Web Services Edge conference by offering 3 full-day training lectures for experienced software developers, architects, and engineers.

Java University℠ Program Take-Aways:
- Training designed and presented by expert Java technology engineers
- Student guides full of source code, examples, references and copies of instructors' materials
- Free Web-based training courses from Sun Educational Services

Tuesday, March 18, 2003
Web Services Using Java™ Technology and XML

SANG SHIN,
SUN MICROSYSTEMS, INC.

Who Should Attend
Web services designers and programmers, application developers, and programmers using the Java programming language who have experience using the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites
Experience using the Java programming language and basic knowledge of XML

Overview
This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

The session will start with an introduction on fundamental concepts and characteristics of Web services. This will be followed by a detailed explanation of how to implement, describe, register, discover, and invoke Web services using core Web services standards - Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI). In addition, the ebXML standard, which defines the framework for the global electronic marketplace will be talked about in detail. Also, the tools for building and deploying Web services will be discussed. Each topic will be presented with concrete examples and demonstrations when possible.

Attendees will also learn how to use standard Java APIs for Web services, mainly Java API for XML Messaging (JAXM), Java technology API for XML-based RPC (JAX-RPC), and Java technology API for XML Registries (JAXR) for developing and deploying Web services.

Benefits
- Learn the fundamental concepts and characteristics of Web services. Gain detailed understanding on core Web services standards: SOAP, WSDL, UDDI.
- Gain a detailed understanding of ebXML, the standard framework for electronic business.
- Learn Java programming language APIs for Web services – JAXM, JAX-RPC, JAXR

Wednesday, March 19, 2003
Java™ 2 Platform: Programmer Certification Fast Path

PHILIP HELLER, PRESIDENT,
HELLER ASSOCIATES

Who Should Attend
This session is designed for programmers who have some exposure to the Java™ programming language, and are ready to prepare for the Sun Certified Programmer for Java 2 Platform exam.

Prerequisites
Object-oriented software development experience and familiarity with the syntax and structure of Java technology-based development.

Overview
The development community recognizes that competency developing solutions using Java technology is vital to productivity, reaffirms your value to your organization, and increases your career advancement opportunities. This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

Benefits
- Receive an intensive review of the advanced topics covered on the Sun Certified Programmer for the Java 2 Platform Exam
- Increase your understanding and knowledge of Java programming language syntax and structure
- Prepare for the exam by reviewing practice tests and questions
- Gain a strong understanding of Java fundamentals

Thursday, March 20, 2003
Java™ 2 Platform: Architect Certification Fast Path

SIMON ROBERTS, TECHNOLOGY
EXPERT AND COURSE DEVELOPER,
SUN MICROSYSTEMS, INC.

Who Should Attend
This session is designed for enterprise application architects, system analysts, experienced technologists, and developers using Java™ technology seeking certification as an architect for the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites
Understand the benefits of Java technology solutions; experience with object-oriented analysis and design; familiarity with concepts of distributed computing.

Overview
Many of the solutions in today's "Net economy" are, or soon will be, developed using the Java 2 Platform, Enterprise Edition (J2EE) architecture. Gaining recognized competency architecting J2EE platform-based solutions is vital to your success as an architect, reaffirms your value, and increases your career opportunities.

This intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives. Multiple real-world case studies are used to demonstrate correctly architected J2EE technology–based solutions and pinpoint key topics presented within the architect exam.

Additionally, you will learn how to interpret exam objectives, what each of the three exam phases contains, and clear guidelines and resources to use after the course.

Benefits
- Receive an intensive review of the topics covered on the Sun Certified Enterprise Architect for the Java 2 Platform, Enterprise Edition Exam
- Increase your understanding and knowledge of successfully architecting solutions using J2EE technology
- Understand the system qualities: scalability, availability, extensibility, performance, and security
- Understand trade-offs of different architectural choices as they pertain to system qualities.
- Describe the benefits and weaknesses of potential J2EE technology–based architectures
- State benefits and costs of persistence management strategies
- Review real-world case studies of J2EE technology–based architecture
- Prepare for the exam by reviewing practice tests and questions

XML Certified Developer *Fast Path*

Audience

This tutorial is for programmers who have some knowledge of XML and related technologies and would like to pass the IBM Certified Developer Test 141 on XML and Related Technologies.

Prerequisites

Background in object-oriented programming and knowledge of Hypertext Markup Language (HTML). Exposure to XML and related technologies.

Overview

XML is the foundation of two important emerging technologies: Web services and the Semantic Web. XML expertise and certification is critical for developers who want to remain competitive in the current tight IT job market. The practice tests and questions in this course are specially designed to teach you XML essentials and the key concepts to successfully pass IBM® Test 141 on XML and related technologies.

Outline

• Well formed XML documents
• XML Infoset
• XML namespaces
• Document analysis and modeling
• Document Type Definitions (DTDs)
• XML Schemas
• The SAX API
• The DOM API
• XPath and XSLT
• XSL Formatting Objects (XSL FOs)
• Formatting XML with CSS
• XLink and XPointer
• XML Encryption
• XML Signatures
• SOAP, UDDI, and WSDL
• XML architectures based on business and technical considerations
• Optimization and testing of XML applications

Presenter Bio

Joel Amoussou is the founder and chief learning architect of XMLMentor. Joel is the author of the first XML training course specially designed to prepare developers for IBM® Test 141 on XML and related technologies. He has created XML content management applications for the aerospace, pharmaceutical, and publishing industries.

Microsoft® FREE .NET Web Services Tutorial

Russ' Tool Shed

Wednesday, March 19, 2003
9:00 a.m. – 5:00 p.m.
Join Russ as he shows you how to use Visual Studio .NET

9-12:15 Intro to Web Services Using VS.NET by Russ Fustino

One of the key ideas behind the .NET strategy is the concept of software as a service, or in short, Web services. This session will explain what a Web service is and provide an overview of its related technologies like XML, SOAP and UDDI. We will demonstrate how the .NET Framework makes it easy to implement them for new and existing applications. This session will also provide concrete best practices for building XML Web services using Visual Studio .NET. We'll answer many common questions like: How will my Web service scale? How can my XML Web services enable interoperability with Web services from other vendors as well as within my own organization? We'll delve into building highly reliable and secure Web services. Also, we will discuss issues such as dealing with complex data types using WSDL (Web Services Description Language), as well as securing SOAP messages using encryption. We'll see how developers can use enterprise-level XML Web services to simplify customer solutions.

1-2:30 - Advanced Web Services Using ASP.NET by Thom Robbins

This session will explore some of the more advanced areas of SOAP in ASP.NET's support for Web services. ASP.NET Web services are the preferred way for Web developers to expose Web services on the Internet. The goal is quick, easy, and high-performing SOAP services. We will look at how to use the SOAP extension classes to create some very interesting applications on top of the core SOAP architecture found within the .NET Framework. For instance, you can implement an encryption algorithm or screen scraping on top of the Web service call. We'll dig into more advanced topics, explore the SOAP headers, and see ways to ensure security in our Web services.

2:45-4:15 - .NET Remoting Essentials by Thom Robbins

Microsoft .NET Remoting is the .NET technology that allows you to easily and quickly build distributed applications. All of the application components can be on one computer or they can be on multiple computers around the world. .NET Remoting allows client applications to use objects in other processes on the same computer or on any other computer to which it can connect over its network. During this presentation we will discuss what you will need to know to get started with .NET Remoting. We will talk about how .NET Remoting compares with DCOM, how to host remoted objects in a variety of applications, how to call remoted objects from a client application, how to control the life time of remoted objects, and how to secure remoting applications.

To learn more, visit
www.sys-con.com

Mobile .NET

Overview

In this session, Derek Ferguson, editor-in-chief of *.NET Developer's Journal*, will give you a thorough introduction to the use of .NET with all manner of mobile computing devices.

The morning will begin with a comprehensive survey of the five most popular mobile computing platforms: Pocket PC, Palm OS, WAP, i-Mode, and J2ME. It will conclude with a thorough examination of how the Mobile Internet Toolkit (a.k.a. "the MIT"), a key part of Microsoft's mobile .NET strategy, can be leveraged to build Web-based applications capable of working with any of these devices.

In the afternoon, we will discuss Microsoft's technology for building self-contained .NET applications for execution on "smart devices" the .NET Compact Framework. Before the end of the session, such advanced topics as COM Interoperability, SQL Server CE, and MIT extensibility will be covered.

Presenter Bio

Derek Ferguson is chief technology evangelist for Expand Beyond Corporation (www.xb.com), the world-wide leader in mobile software for enterprise management. He is also editor-in-chief of *.NET Developer's Journal* and author of the book *Mobile .NET*.

Derek Ferguson
Editor-in-Chief, *.NET Developer's Journal*
Chief Technology Evangelist, Expand Beyond Corp.

Register by Phone 201 802-3058

Major Sponsors at the World's Largest Web Services, Java, XML, and .NET Conference and Expo

2003 East Gold Sponsors

COMPUWARE.

Compuware will be the best worldwide provider of quality software products and services designed to increase productivity. We will continue to create practical solutions that meet our customers needs and surpass their expectations. We will provide an environment for our employees where excellence is encouraged and rewarded and where diversity is promoted at all levels of the company. Compuware is built upon a diverse employee population with a set of key attributes that is common to every individual: Secure, Moral, Ethical, In Agreement with Economic Purpose, Team Player, Other Oriented, Positive, Optimistic, Energetic Achiever and Bright.

Microsoft®

Introducing Visual Studio .NET; visionary yet practical, the single comprehensive development tool for creating the next generation of applications has arrived. Developers can use Visual Studio .NET to:
• Build the next-generation Internet.
• Create powerful applications fast and effectively.
• Span any platform or device.
Visual Studio .NET is the only development environment built from the ground up for XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

ORACLE®

Oracle Corporation (Nasdaq: ORCL) is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than $9.4 billion, the company offers its database, tools, and application products, along with related consulting, education, and support services. Headquartered in Redwood Shores, California, Oracle is the first software company to develop and deploy 100 percent Internet-enabled enterprise software across its entire product line: database, server, enterprise business applications, and application development, and decision support tools.

2003 East Silver Sponsor

hp invent

HP is a leading provider of products, technologies, solutions and services to consumers and business. The company's offerings span IT infrastructure, personal computing and access devices, global services, and imaging and printing. Our $4 billion (U.S.) annual R&D investment fuels the invention of products, solutions and new technologies, so that we can better serve customers and enter new markets. We invent, engineer, and deliver technology solutions that drive business value, create social value, and improve the lives of our customers.

2003 East Featured Sponsor

Sun microsystems

Sun was founded with one driving vision. A vision of computers that talk to each other no matter who built them. A vision in which technology works for you, not the other way around. While others protected proprietary, stand-alone architectures, we focused on taking companies into the network age, providing systems and software with the scalability and reliability needed to drive the electronic marketplace.

2003 East Corporate Sponsors

IONA

IONA is the leading e-Business Platform provider for Web Services Integration. IONA Orbix E2A enables the flow of information across disparate systems and the liberation of application assets for new business opportunities. For additional information about IONA, visit our Web site at http://www.iona.com.

Rational®
the e-development company™

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life-cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 rely on Rational tools and services to build better software, faster. Additional information is available on the Internet at http://www.rational.com

SAMS

Sams Publishing has more than 500 titles in print, and is one of the most successful computer book publishers in the world. From introductory tutorials to comprehensive reference books, Sams Publishing focuses on teaching tomorrow's programmers, developers and system administrators the skills they need to build and manage emerging technologies.

SYBASE

Sybase has always delivered solutions that help customers to share data, is platform independent, and integrates everything: platforms, application servers, components, databases, portals, processes, message brokers, and mobile/wireless. Our technologies have the insight that will promote ease of use, leverage best practices, ensure positive ROI, and help your organization build a successful, pragmatic strategy based on next-generation technologies. Check us out at http://www.sybase.com/solutions/e-business

DIGITAL EVOLUTION
The Business Value of Web Services

DE solutions and thought leadership deliver the value of Web services to business users today. DE unleashes the value of underlying IT assets, giving enterprises the freedom to innovate and evolve business processes within a standards-based, vendor neutral, loosely coupled framework. Some of the largest corporations in the world rely on DE's Web services solutions. www.digev.com

The Largest Expo Focusing on Java, Web Services, .NET & XML

Walk the Expo Floor, visit each of these pioneers, and get the answers you need!

Please refer to the conference Web site for an updated list of sponsors & exhibitors

Expo Hours

Exhibit Hall A
Wednesday, March 19:
11:00am - 6:00pm
Thursday, March 20:
11:00am - 4:00pm

Opening Night Reception on the Expo Floor
Wednesday, March 19:
5:00pm - 6:00pm

- Actional
- Altio
- Altova
- asp.netPRO
- ASPstreet.com
- Attachmate
- BEA WebLogic Developer's Journal
- Borland
- ColdFusion Developer's Journal
- Compuware
- Digital Evolution
- EAI Industry Consortium
- Ektron
- Fair, Isaac and Company
- Forum Systems
- GAO Web Services Inc.
- Hewlett-Packard
- HSPstreet.com
- Infragistics
- IONA

- Java Developer's Journal
- JavaWorld
- McCabe & Associates
- Melissa Data
- Merant
- Microsoft
- Mindreef
- .NET Developer's Journal
- Novell
- OASIS
- ParaSoft
- PerfectXML
- Rational Software
- SAMS Publishing
- SD Times
- SoftArtisans
- Software AG
- Sonic Software
- SpiritSoft
- Sun Microsystems

- Sybase
- TeamStudio
- TogetherSoft
- Web Services Journal
- WebSphere® Developer's Journal
- Wireless Business & Technology
- XML-Journal
- Xtremesoft

(List as of December 2002)

HOTEL ARRANGEMENTS ARE EASIER THAN EVER!

Special arrangements have been made with some of Boston's finest hotels, priced well below regular rates. Hotels are located within walking distance, if not connected to the Hynes Convention Center. To learn more about hotel savings, call ETI at (800) 829-2281 or (201) 444-0060 (direct) or fax reservations to (201) 444-0062. To make online reservations visit www.expotravel.com by February 24, 2003.

SPECIAL RATES

Official Hotel	Address	Single	Double
Hilton Boston Back Bay	40 Dalton Street	$149	$149
Sheraton Boston Hotel	39 Dalton Street	$159	$159

To make reservations call (800) 829-2281or (201) 444-0060 (direct). Fax reservations to (201) 444-0062. Credit card information is required to guarantee reservations and expedite confirmation. Confirmations will be mailed directly from the hotel, time permitting. All changes and cancellations should be made directly through ETI.

Special Arrangements have been made with these Airlines for Web Services Edge East 2003 Attendees!

American Airlines is proud to offer discounted fares to Web Services Edge East 2003 attendees to fly to Boston this March 18-20, 2003.
Valid travel dates are March 13 to March 25, 2003.
American is offering 5% off the lowest published fare or 10% off full Coach, Business and First class fares. In addition, American is extending a 5% bonus discount for ticketing 30 days in advance. For reservations, call American at (800) 433-1790 and reference STAR file number A5433AW

Continental Airlines is proud to offer Web Services Edge East 2003 attendees Special Fares! Fly to Boston & Web Services Edge East 2003 this March 18-20, 2003.
Special rates are valid for travel from March 13 to March 25, 2003.

For Reservations & Ticketing:
Contact Continental MeetingWorks (6:00am - 10:00pm CST , Seven (7) day a week) at (800) 468-7022 (United States or Canada) and provide the Z Code ZWMF and Agreement Code VSD04D. For International reservations contact the local Continental Airlines Reservations Office.

AVIS is Proud to Offer Special Rates for:
Web Services Edge East 2003
March 18 - 20, 2003
Boston, MA
Avis Worldwide Discount (AWD) Number: D918596
Call (800) 331-1600

Register by Phone 201 802-3058

REGISTRATION FORM

CONFERENCE: March 18 – 20, 2003   EXPO: March 19 – 20, 2003

John B. Hynes Veteran Memorial Convention Center • Boston, MA

THREE WAYS TO REGISTER FOR CONFERENCE

1) On the Web: Credit Cards or "Bill Me" Please make checks payable to SYS-CON Events
2) By Fax: Credit Cards or "Bill Me" 201-782-9651
3) By Mail: 135 Chestnut Ridge Road, Montvale, New Jersey 07645, Attention: Registration

Please note: Registrations are not confirmed until payment is received.

Please complete sections 1, 2, 3 and 4

1 **YOUR INFORMATION** (Please Print)   ☐ Mr.   ☐ Ms.

First Name _____ Last Name _____
Title _____
Company _____
Street _____
Mail Stop _____
City _____
State_____ Zip _____ Country _____
Phone _____
Fax _____ E-Mail _____

2 **PAYMENT METHOD:** (Payment in full due with registration)

☐ Check or Money Order Enclosed (Registration confirmed upon receipt of payment)

Check #_____ Amount of Check $ _____

Charge my   ☐ Visa   ☐ MasterCard   ☐ American Express   ☐ Discover

Name on card_____

Card #_____ Exp. Date_____

Signature_____

Billing Address (if different from mailing address)
_____

3 **PLEASE INDICATE YOUR CONFERENCE CHOICE**

Total Registration fee $_____

	Before 2/28/03	Before 3/14/03	On Site
☐ **GP Gold Passport** Good for all three days of the .NET, Web Services, XML, Java, and Vendor Tracks, including the XML Certified Developer Fast Path Tutorial, Keynotes, Panel Discussions, preferred seating for Microsoft .NET's Russ' "Tool Shed" Tutorial, and your choice of One Sun Microsystems Java˙ University Class	$1,495.00	$1,695.00	$1,795.00
☐ **3D Three Day Conference** (Does not include Sun Java™ Education)	$1,295.00	$1,495.00	$1,695.00
☐ **2 Day Conference** (Does not include Sun Java™ Education) (select any two days: ☐ Tue. ☐ Wed. ☐ Thurs.)	$1,195.00	$1,295.00	$1,395.00
☐ **1 Day Conference** (Does not include Sun Java™ Education) (select any one day: ☐ Tue. ☐ Wed. ☐ Thurs.)	$495.00	$695.00	$895.00
☐ **JU1 Sun Java™ University Class** Select one: ☐ Web Services Programming Using Java™ Technology and XML (Mar.18) ☐ Java™ Fast Path: Programmer (Mar.19) ☐ Java™ Fast Path: Architect (Mar. 20)	$695.00	$895.00	$995.00
☐ **JU2 Sun Java™ University Class** Select two: ☐ Web Services Programming Using Java™ Technology and XML (Mar.18) ☐ Java™ Fast Path: Programmer (Mar.19) ☐ Java™ Fast Path: Architect (Mar. 20)	$1,295.00	$1,495.00	$1,595.00
☐ **JU3 Sun Java™ University Class** Select three: ☐ Web Services Programming Using Java™ Technology and XML (Mar.18) ☐ Java™ Fast Path: Programmer (Mar.19) ☐ Java™ Fast Path: Architect (Mar. 20)	$1,495.00	$1,695.00	$1,895.00

4

A. Your Job Title
☐ CTO, CIO, VP, Chief Architect
☐ Software Development Director/Manager/Evangelist
☐ IT Director/Manager
☐ Project Manager/Project Leader/Group Leader
☐ Software Architect/Systems Analyst
☐ Application Programmer/Evangelist
☐ Database Administrator/Programmer
☐ Software Developer/Systems Integrator/Consultant
☐ Web Programmers
☐ CEO/COO/President/Chairman/Owner/Partner
☐ VP/Director/Manager Marketing, Sales
☐ VP/Director/Manager of Product Development
☐ General Division Manager/Department Manager
☐ Other (please specify)_____

B. Business/Industry
☐ Computer Software
☐ Computer Hardware and Electronics
☐ Computer Networking & Telecommunications
☐ Internet/Web/E-commerce
☐ Consulting & Systems Integrator
☐ Financial Services
☐ Manufacturing
☐ Wholesale/Retail/Distribution
☐ Transportation
☐ Travel/Hospitality
☐ Government/Military/Aerospace
☐ Health Care/Medical
☐ Insurance/Legal
☐ Education
☐ Utilities
☐ Architecture/Construction/Real Estate
☐ Agriculture
☐ Nonprofit/Religious
☐ Other (please specify) _____

C. Total Number of Employees at Your Location and Entire Organization (check all that apply):

	Location	Company
10,000 or more	01 ☐	01 ☐
5,000 - 9,999	02 ☐	02 ☐
1,000 - 4,999	03 ☐	03 ☐
500 - 999	04 ☐	04 ☐
100-499	05 ☐	05 ☐
100 or less	06 ☐	06 ☐

D. Please indicate the value of communications and computer products and services that you recommend, buy, specify or approve over the course of one year:

☐ $10 million or more
☐ $1 million - $9.9 million
☐ $500,000 - $999,999
☐ $100,000 - $499,999
☐ $10,000 - $99,999
☐ Less than $10,000
☐ Don't know

E. What is your company's gross annual revenue?
☐ $10 billion or more
☐ $1 billion - $9.9 billion
☐ $100 million - $999 million
☐ $10 million - $99.9 million
☐ $1 million - $9.9 million
☐ Less than $1 million
☐ Don't know

F. Do you recommend, specify, evaluate, approve or purchase wireless products or services for your organization?
01 ☐ Yes   02   ☐ No

G. Which of the following products, services, and/or technologies do you currently approve, specify or recommend the purchase of?
☐ Application Servers
☐ Web Servers
☐ Server Side Hardware
☐ Client Side Hardware
☐ Wireless Device Hardware
☐ Databases
☐ Java IDEs
☐ Class Libraries
☐ Software Testing Tools
☐ Web Testing Tools
☐ Modeling Tools
☐ Team Development Tools
☐ Installation Tools
☐ Frameworks
☐ Database Access Tools / JDBC Devices
☐ Application Integration Tools
☐ Enterprise Development Tool Suites
☐ Messaging Tools
☐ Reporting Tools
☐ Debugging Tools
☐ Virtual Machines
☐ Wireless Development Tools
☐ XML Tools
☐ Web Services Development Toolkits
☐ Professional Training Services
☐ Other [Please Specify] _____

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3058 by March 4, 2003.

SYS-CON EVENTS

Special Insert: Web Services Edge East Conference & Expo
Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Register Online @ sys-con.com

CTIA
Wireless 2003

www.ctiashow.com

WRITTEN BY **T.V. RAMAN**

Collecting Business Critical Information

Changing the paradigm

Since their inception in 1993, HTML forms have come to be the underpinnings of user interaction on the World Wide Web. The convenience afforded by the ability to provide consistent end-user access to information and application services via a universal Web browser created a platform-independent environment for electronic commerce across the Internet. However, building on the essential simplicity of HTML forms has resulted in an extremely complex Web programming model.

Today, Web application developers are forced to work at different levels of abstraction to deliver a satisfactory end-user experience. Given the need to perform electronic transactions with a variety of different devices and user-interface modalities, the problems become even more complex. As we deploy Web access to software at all levels of complexity, ranging from business back ends to simple electronic transactions, we can better address these issues by revisiting the design of HTML forms that are the essential underpinnings of the transactional Web.

XForms 1.0 is a revision to the existing HTML forms technology developed by the W3C and builds on the advantages of XML to create a versatile forms module that can stand the Web in good stead for the next decade. XForms provides an interoperable forms solution based on XML that can significantly ease the way structured information is collected and shared across the Internet. In conjunction with XML-based Web services, XForms goes the final step in connecting end users to their business data.

Leveraging XML for Collecting Information

Business workflows are about collecting and categorizing information that starts off in semistructured documents, for example, a travel expense report, and eventually moving the fully structured information into the relevant back-end information processing system. Until now, structured data has been the purview of database applications, while semistructured documents have been managed with a variety of tools, including word processors and spreadsheet programs. With the advent of the Web, such data gathering has also come to include Web interfaces that are typically deployed to a universal Web client.

Though this move to online interaction has brought significant productivity gains, businesses and end users still fail to reap the full benefits of electronic documents because of the need to move data across disparate systems. XML promises to bridge this gap by enabling the interchange of structured and semistructured information. In this context, XForms enables the creation of browser-based interfaces for editing and submitting XML documents. By enabling the collection and validation of structured data within a Web browser, XForms makes the original promise of the document as the interface a reality; users can now interact with Web documents that present a semistructured document-based view of the underlying structured data.

A key consequence of this evolution is that data collected via such interfaces will no longer need to be reentered into the relevant back-end information system; instead, the collected XML data can be submitted directly to a Web service that connects the user with the underlying information technology. Thus, XForms looks and feels to the end user like today's Web forms enhanced with all of the data-capture functionality you'd expect from a traditional forms package. In addition, it has been designed to be suitable for delivery to the emerging variety of pervasive devices ranging from PDAs to smart phones. Built from the ground up to work with XML, XForms can gather information that has been structured to conform to the customer's document schema, where the structure and type of content that each data element can contain has been designed to suit the needs of a specific business application. The XML instance that results from a user interacting with such a form can then be integrated with existing databases and servers, making it easier to reuse data across the enterprise via XML Web services.

XForms Innovations

One of the key innovations introduced by XForms is the separation of the data being collected from the interaction that is presented to the user. This separation of the model from the view is a key enabler with respect to creating document-based user interfaces that collect and validate structured data. In this context, XForms is designed to be embedded within other XML document formats, such as XHTML. When embedded in XHTML, the XForms model defines regions of structured data within the overall XHTML document, which continues as a traditional semistructured document. XForms user interface controls embedded within the XHTML markup connect to the underlying XForms model, thereby enabling the display and update of the values stored in the XForms instance. Finally, the data

AUTHOR BIO

T.V. Raman is an accomplished computer scientist, with more than eight years of industry experience in advanced technology development. He has authored 2 books and filed over 20 patents. T.V. participates in numerous W3C working groups and authored Aural CSS (ACSS); in 1996 he wrote the first ACSS implementation. He has been actively participating in defining XML specifications for the next-generation Web including XForms, XML Events, and XHTML+Voice.

collected by the XForms application can be cleanly extracted and serialized as a structured XML document for submission to a Web service. XForms can thus address the continuum ranging from fully structured documents such as database queries to semistructured documents like expense reports, project reviews, and travel itineraries. XForms documents, when deployed inside a Web browser, provide the best of both worlds; they provide the ease of use and platform independence provided by Web forms, and the rigorous data-capture capabilities of traditional forms packages. The end result is that users end up creating structured information where they would have earlier created completely unstructured content within a myriad of desktop applications, all having their own proprietary data formats that would have required reentering the data in order to get the content ready for processing by the underlying information technology.

A key consequence of this evolution is that information technologists can continue to model business data using abstract structures that are amenable to machine processing. XForms binds a user-friendly Web browser interface to such abstract XML models, thereby empowering the average user to edit and update these abstract structures. In this sense, XForms enables a standard Web browser to associate editable views to the underlying abstract XML models. This ability to view and edit XML documents from within a standard Web browser is likely to prove a key empowering technology. In addition, industry-standard XML Schema validation and business logic validation through the XForms model enables error checking at each stage, thereby avoiding costly data errors. Finally, the XForms architecture is designed to enable the use of off-line forms, i.e. users can save forms to their computers and work on them when not connected.

Dynamic Multimodal Interaction

In addition to the static type checking provided by XML Schema, XForms includes the ability to express dynamic runtime constraints that can be used in creating smart user interfaces that react to the information collected at any given point of the user interaction. Thus, a complex insurance form might include a section for each of the applicant's children. Static constraints that are expressible via XML Schema are used to specify the type and structure of each of these sections. At runtime, the form needs to be updated dynamically based on the number of children the applicant has; thus, if the applicant has no children, all of the sections of the form pertaining to the applicant's children need to be made inactive. The XForms processing model enables the creation of such dynamic user interfaces by allowing authors to express dynamic constraints within the XForms model that further refine the constraints expressed using XML Schema. Support for such dynamic interaction is enhanced via XForms user interface constructs that allow the author to create template-based user interfaces. As an example, authors can use the XForms construct <repeat> to create user interfaces that iterate over the items of a collection. This functionality can be used to advantage in producing user interfaces that allow the user to interact with structures that grow or shrink dynamically. Thus, a purchase order form might start with the user interface needed for ordering a small number of items, but can grow to accommodate additional orders. XForms includes support for a rich set of declarative action handlers that in conjunction with XML Events can create rich user interface widgets such as dynamic wizards.

The views bound to the underlying XForms model may be different visual views, e.g. a complex form that binds a details view and a summary view to the underlying data. Binding to a single XForms model automatically synchronizes these views, thereby enabling the user to view an updated summary as user interaction progresses. With the coming of age of speech interaction, and the ability to integrate speech interaction into XHTML, these views can be more than just visual views, one can bind visual and auditory view to the same underlying XForms model to produce rich multimodal interaction.

Conclusion

Online transactions form the core of most Web applications, and electronic forms are intrinsic to such transactions. Electronic transactions collect data from the end user, transmit the collected data to the appropriate back-end information technology, and communicate the result of the transaction to the user. With the advent of Web services, the data that is collected is submitted to the information back end as a structured XML document.

In this context, connecting the end user to the actual information requires the creation of intuitive user interfaces for editing and updating such structured XML documents. Deploying such user interfaces within a universal Web browser has enormous advantages for reducing the cost of deploying online transactions. With the advent of an increasing variety of electronic devices, it becomes critical to be able to perform such transactions using a variety of access devices and modalities. XForms provides an interoperable means for deploying uni-

> ## "Built from the ground up to work with XML, XForms can gather information that has been structured to conform to the customer's document schema"

versal access to XML content by enabling the creation and editing of structured XML content from within a universal Web client.

Acknowledgments

References

- *XForms 1.0:* www.w3.org/TR/xforms
- *HTML:* www.w3.org/TR/html4
- *XML:* www.w3.org/TR/rec-xml
- *XML Events:* www.w3.org/TR/xml-events
- *Speech Interaction:* www.w3.org/Voice
- *XHTML:* www.w3.org/TR/xhtml%2Bvoice
- *Rich Multimodal Interaction:* www.w3.org/2002/mmi

TVRAMAN@ALMADEN.IBM.COM

WRITTEN BY **EUGENE KUZNETSOV**

The Adoption:Hype Ratio

A pragmatic look at XML Web services deployments

Even today, I'm frequently faced with those who are disappointed at the progress of XML and Web services. My stock answer to skeptical questions about XML adoption is to talk instead about the "Adoption:Hype" ratio and agree that its value is at or near zero. The many successful projects cannot possibly match the infinite hype associated with the technologies.

Often, the enthusiasm of even the most technical groups outpaces what can actually be achieved in production given the readiness of the software packages, required standards, corporate politics, and developer tools. What is needed is a pragmatic approach to XML and Web services, one that reconciles both that real mission-critical projects are being put into production today and that some parts of the industry are not yet ready.

Reasoning About Performance

In an ideal world, networks are infinitely fast, bandwidth is free, and there is no overhead to using XML messages or SOAP calls to integrate applications. The real-world constraints can be easily analyzed with some simple rules. First, calculate both the total network bandwidth consumption and processing speed of a given transaction. Network overhead in megabits can be approximated by: FILESIZE + HEADERSIZE (KB) * 0.008. Each transaction will have a fixed overhead due to various headers (e.g., IP, HTTP, SOAP), usually insignificant if HEADERSIZE is much smaller than FILESIZE. For example, let's take a simple dynamic XML Web site. If your new Web site uses 50KB XML input files, transforms them using XSLT, and produces 100KB of HTML as output, assuming HTTP overhead of

0.50KB per transaction, each transaction will consume approximately 1.2 megabits. Therefore, the maximum possible TPS = (100MBps/1.2MB) on a fast ethernet network will be somewhere around 83 transactions per second (TPS). (*Note:* This makes several simplifying assumptions, including a half-duplex network that has no other traffic.)

Second, calculate the effect of XML processing speed. With common software engines, XML parsing throughput ranges between 4 and 40MBps, and XSLT is between 1 and 10MBps. Schema validation, complexity of XPath/XSLT processing, and the greater node-per-byte ratio of document content all serve to reduce the throughput. Because the ranges are wide and often depend on both hardware and software configuration, it's best to use a standard tool (such as XSLTMark or SOAPMark, see Table 1) to find out how long a specific application processing step takes. For this example, let's assume a total throughput of about 10MBps or 120ms per complete transaction. This would mean a best-case maximum of only 8.3 TPS per CPU, much lower than the wirespeed limit of 83 TPS calculated earlier. At least 10 CPUs scaling in an ideal fashion would be required to reach the 83 TPS limit. Similarly, 1 GB/sec speed would require 100 CPUs, unless special XML accelerators are employed.

This kind of first-order reasoning about throughput is very helpful when discussing business requirements for XML network applications. You'll find it just as helpful in sanity-checking figures from the first performance lab tests as in applying common sense to unreasonably high TPS requirements handed down from above.

Real-World Security

Another area of considerable practical difficulty is security, especially when it comes to Web services. While the security issue is finally receiving some much-needed attention, there is not enough clarity to be useful in practice. There is a misconception that no security is needed inside the firewall, and any externally exposed Web service must necessarily use the complete set of security specifications. Neither is true in practice: the enterprise security perimeter is increasingly "dissolving," to use a term originated by @stake (a prominent security consulting firm), and there are ways to deploy secure XML Web services today, without waiting for all the specifications to become fully baked.

There's a pragmatic approach that allows applications to be deployed today and to migrate to the latest specifications step-by-step. It's motivated by the belief that internal threats are no less dangerous than external ones, and that today's internal Web services project may need to be made available to external partners in a rush tomorrow. The first step is securing the transport layer using existing encryption technology. Although IPSec-based VPNs are the technology traditionally associated with securing extranets, SSL has proven easier to deploy and provides a more flexible security model. It is easily deployed between servers that are both inches apart and thousands of miles apart. Primitive and oft-cumbersome but secure access control can be implemented by using SSL client certificates. Other examples of pragmatically adapting existing security technologies or concepts to the world of Web services security are secure transaction logging, PKI integration, and CRL.

Another simple but essential tech-

AUTHOR BIO

Eugene Kuznetsov, founder, president, and CTO of Datapower Technology, Inc., founded DataPower Technology in 1999 to provide XML-aware network infrastructure for enterprise application development. Prior to starting DataPower, Eugene led the Java JIT Compiler effort for Microsoft Corporation's support for Internet Explorer for Macintosh. Eugene has been a featured speaker at several industry conferences, including Networld+Interop and Web Services DevCon, and he is a serial author. Eugene holds a BS in electrical engineering from the Massachusetts Institute of Technology.

Test Case	Input Size	Input Description	Stylesheet	Notes
alphabetize	medium	100-row database table	select, control	Sorts the input tree according to element name.
attsets	small	sales report	general	Tests node-copying using named attribute sets.
avts	medium	100-row database table	select	Tests attribute-value template expansion.
axis	small		select	Tests XPath selection along the different axes.
backwards	small		control	Reverses order of elements in input document.
bottles	small	initial size parameter	function, control	Generates "99 bottles of beer on the wall" song.

Table 1 • XSLTMark cutaway example

nique is hiding the internals from the "outside," at both the network and application layers. For networks, this is done using familiar NAT (Network Address Translation) and/or proxying. At the next level up the stack, the HTTP protocol layer, URL rewriting serves to obscure endpoints and their structure. Finally, at the XML application layer, transforming all the XML messages (or requests and responses) provides what can be thought of as "XML Address Translation": mapping between the private internal data layout and the external one. This kind of application-layer protection is easily implemented today using XSLT, one of the most mature XML technologies.

The fluid state of many Web services security specifications doesn't have to be an impediment to using some of them today. XML Signature and XML Encryption have reached REC status in W3C and can be deployed without risk of future changes. However, incompatibilities between different implementations are still frequent, and practical experience suggests that careful interoperability testing should be conducted on all software and/or hardware used for XML crypto processing. For newer specifications, choose XML security packages or solutions that can be easily adapted to changes in specifications. The great thing about XML security standards is that they uniformly use XML as the data-encoding format, so their implementations should possess flexibility comparable to that of all other XML-handling software. The assumption should be that all the standards will experience minor changes in the next six months – the software should be designed with this in mind to avoid quick obsolescence.

Although standards bodies are only now beginning to put minimum performance requirements into specifications, an important practical consideration is the performance impact of any

added security. Securing XML Web services is largely an XML-processing problem, so performance is again an important issue in practice (see Figure 1). For widely available XML engines, a simple rule of thumb is that schema validation and parsing is 3–4 times more time-consuming that parsing alone. Even the performance of XML Encryption and XML Digital Signature turns out to be dominated by XML processing operations.

One estimate puts the processing overhead of a fully secured XML trans-

ing as well. Remember that well-encrypted data cannot be compressed, so compression will be less effective for reducing the network impact of transactions with a lot of encrypted data.

It's also important not to forget about the deployment and operational aspects of adding security capability. Today, XML Web services security is an issue that sits between two separate organizations, the applications group and the security group. This can lead to great difficulty in getting apps into production,
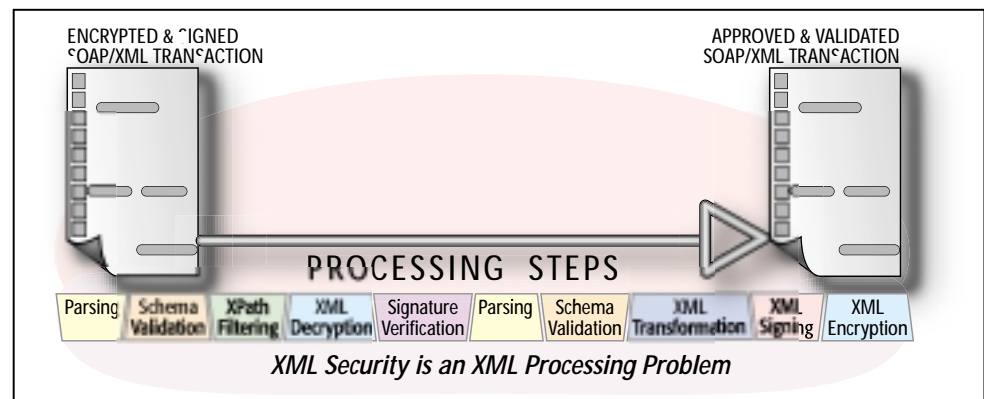

Figure 1 • XML processing

action at 10 times that of a one without any security features. So it's not surprising that there are Web services deployments today that knowingly disable essential security functions, such as schema validation and message transformation, to get the required performance. This behavior causes much consternation among security analysts, but is the kind of security-or-performance choice that's familiar to those actually implementing XML applications. The reality is that much like overly complex home burglar alarms and metal detectors with lengthy lines, any Web services security function that slows down transactions is often left turned-off. The approaches for performance planning mentioned above can be used for estimating the impact of security process-

and is again a practical (although not at all technical) issue that's important to be aware of.

Proof Is in the Practice

This article attempted a brief overview of large-scale XML and Web services deployment from a practical perspective. There is much more to be said about Web services deployment and how the model of loosely coupled entities can solve some old practical problems and introduce brand new ones. The important thing is that with the application of common sense to both sides of the Adoption:Hype ratio, XML technology will continue its upward trend. ◈

eugene@datapower.com

Introduction to SOAP Serialization Rules

From *Essential XML Quick Reference*

Essential XML Quick Reference

A Programmer's Reference to XML, XPath, XSLT, XML Schema, SOAP and More
by Aaron Skonnard and Martin Gudgin

Published by: Addison-Wesley Pub Co.
Pages: 402
ISBN: 0201740958
List Price: $24.99

SOAP defines a set of serialization rules for encoding datatypes in XML. All data is serialized as elements rather than attributes. Attributes are only used for structural metadata; for example, when references are needed.

For simple types such as strings, numbers, dates, and so forth, the datatypes defined in "XML Schema Part II—Datatypes" are used. For types such as classes or structures, each field in the type is serialized using an element with the same name as the field. For array types, each array element is typically serialized using an element with the same name as the type, although other element names may be used. In both cases, if the field being serialized is itself a structure or an array, then nested elements are used. The top-level element in both the structure case and the array case is namespace qualified. Descendant elements should be unqualified.

The serialization rules apply to children of the Header element as well as children of the Body element. Such children are serialized types just like any other type. A request and any associated response are also treated as types, and are serialized according to the same rules.

Examples

For an example of serialization of a structured Java or VB type, see Listing 1.

For an example of serialization of a Java or VB array, see Listing 2.

10.4.1 Serialization of Simple Structured Data

Serializing data structures, when each field is referred to exactly once, is straightforward. Each field is serialized as an embedded element, a descendant element of the Body element, not as an immediate child. Such an element is called a single-reference accessor, and it provides access to the data in the field at a single location in the message. The element name used to contain the data is the same as the field name used in the programmatic type.

For an example of serializing structured data, see Listing 3, which shows the Java and VB definitions for a method call taking a structured type representing a Person as a single parameter, followed by the SOAP message representing a request to execute such a method.

10.4.2 Serialization of Structured Data with Multiple References

In cases when a field in a data structure is referred to in several places in that data structure (for example, in a doubly linked list), then the field is serialized as an independent element, an immediate child element of Body, and must have an ID attribute of type ID. Such elements are called multireference accessors. They provide access to the data in the field from multiple locations in the message. Each reference to the field in the data structure is serialized as an empty element with an href attribute of type IDREF, where the value of the attribute contains the identifier specified in the ID attribute on the multireference accessor preceded by a fragment identifier, #.

Listing 4 shows the Java definition for a method call taking two parameters both of type Person, followed by the SOAP message representing a request to execute such a method where both parameters refer to the same instance of Person.

10.4.3 Dealing with Null References in Complex Data Structures

In certain cases when reference types exist in a programmatic data structure there is a need to represent a null reference. Such references are modeled in SOAP messages using the nil attribute in the `http://www.w3.org/2001/XMLSchema-instance` namespace. Setting the value of the attribute to 1 indicates that the accessor on which it appears represents a null reference. See Listing 5 is an example of null references, a Java class definition for a simple linked list. The end of the list is indicated by a null reference in the next field. A list of three items is passed in the request message.

10.4.4 Serializing Dynamically Typed Data

SOAP provides for serialization of dynamically typed data; that is, data typed at runtime, through a polymorphic accessor. Such accessors look like normal accessors apart from the presence of a type in the `http://www.w3.org/2001/XMLSchema-instance'` namespace. This attribute indicates the type the accessor actually holds. The value of this attribute may well vary from message to message. Listing 6, an example of dynamically typed data, shows the Java and VB definitions for a method call taking a dynamically typed parameter followed by several SOAP messages representing a request to execute such a method. The first SOAP message passes a parameter of type long whereas the second passes a parameter of type Person.

10.4.5 Arrays

SOAP provides comprehensive array support. Single and multidimensional arrays are supported, along with sparse

and jagged arrays and partial transmission. Arrays in SOAP are always of type Array in the `http://schemas.xml-soap.org/soap/encoding/` namespace, or a type derived by restriction from that type. If they are of the Array type, they are encoded using an Array element also in the `http://schemas.xmlsoap.org/soap/encoding/` namespace. If they are of a derived type, then any element name may be used. In either case, an arrayType attribute in the `http://schemas.xml-soap.org/soap/encoding/` namespace is mandatory. The type of this attribute is string, but it in fact indicates the type of the array along with dimension information. Each dimension appears in square brackets after the QName for the type, separated by commas. Each array item is serialized as an element. The name of this element can be the type name or some arbitrary name. Listing 7, a simple array example, shows a response message containing an array of five long values. Note the value of the arrayType attribute indicating the size of the array.

10.4.6 Multidimensional Arrays

Multidimensional arrays can be encoded by specifying multiple dimensions separated by commas inside the square brackets in the arrayType attribute. Any number of dimensions may be specified. Listing 8, a multidimensional array example, shows a request message containing a two-dimensional array of strings. Note the value of the arrayType attribute indicating the type and dimensions of the array.

10.4.7 Partial Transmission of Arrays

In certain scenarios an array of a certain size may need to be transmitted, but only a subset of the items needs to be sent. For such arrays the array element is annotated with an offset attribute in the `http://schemas.xml-soap.org/soap/encoding/` namespace. The value of the offset attribute indicates the zero-based offset of the first element. The value appears in square brackets. Listed items are assumed to appear at contiguous locations in the array. Items may be omitted from the end of the array. Listing 9, an example of partial array transmission, shows a request message that transmits the third, fourth, and fifth items in a nine-item array

10.4.8 Sparse Arrays

Sparse arrays, those in which noncontiguous items need to be transmitted, are also supported. Each serialized array item is annotated with a position attribute in the `http://schemas.xmlsoap.org/soap/encoding/` namespace. The value of the position attribute is a zero-based offset of the position of the item in the array, enclosed in square brackets. Listing 10, an example of sparse arrays, shows a request message that transmits the second, fourth, and eighth items in a nine-item array.

10.4.9 Jagged Arrays

SOAP supports jagged arrays, also known as arrays of arrays. The arrayType attribute contains a type that includes empty square brackets, as many as necessary to indicate how many dimensions each array has, followed by the dimensions of the array of arrays in square brackets as normal. The inner array elements are also annotated with the appropriate arrayType attribute. Listing 11, an example of jagged arrays with single-reference accessors, shows a request message that transmits an array of arrays of strings. Each array is encoded using a single-reference accessor. Listing 12, an example of jagged arrays with multireference accessors, shows a request message that transmits an array of arrays of strings. Each array is encoded using a multireference accessor. ◆

LISTING 1 • Serialization of a structured Java or VB type

```
package example.org.People;
//Java class definition
class Person
{
String name;
float age;
short height;
}

//VB Type definition
Public Type Person
name As String
age As Single
height As Integer
End Type


<p:Person
xmlns:p='urn:example-org:people'>
<name>Martin</name>
<age>33</age>
<height>64</height>
</p:Person>
```

LISTING 2 • Serialization of a Java or VB array

```
package example.org.Num;
//Java class definition
class Numbers
{
long [5] data;;
}

//VB Type definition
Public Type Numbers
data(5)As Long
End Type

<p:Numbers
xmlns:p='urn:example-org:num'>
<data enc:arrayType='xsd:long [5]'
xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'>
<enc:long>2</enc:long>
<enc:long>3</enc:long>
<enc:long>5</enc:long>
<enc:long>7</enc:long>
<enc:long>9</enc:long>
</data>
</p:Numbers>
```

LISTING 3 • Serializing structured data

```
package example.org.People;
//Java class definitions
class PersonName
{
String givenName;
String familyName;
}

class Person
{
PersonName name;
float age;
short height;

public static void AddPerson (Person person );
}
//VB Type definitions
Public Type PersonName
givenName As String
familyName As String
End Type

Public Type Person
name As PersonName
age As Single
height As Integer
End Type

Public Sub AddPerson (ByRef person As Person )
End Sub

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>
<soap:Body>
<p:AddPerson

xmlns:p='urn:example-org:people'>
<person>
   <name>
```

```
      <givenName>Martin</givenName>
      <familyName>Gudgin</familyName>
    </name>
    <age>33</age>
    <height>64</height>
</person>
</p:AddPerson>
    </soap:Body>
</soap:Envelope>
```

LISTING 4 • Multireference accessors

```
package example.org.People;
//Java class definition
class PersonName
{
String givenName;
String familyName;
}

class Person
{
PersonName name;
float age;
short height;

public static boolean Compare (Person p1,Person p2 );
}

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>
  <soap:Body xmlns:p='urn:example-org:people'>
    <p:Compare>
      <p1 href='#pid1'/>
      <p2 href='#pid1'/>
    </p:Compare>
    <p:Person id='pid1'>
      <name>
        <givenName>Martin</givenName>
        <familyName>Gudgin</familyName>
      </name>
      <age>33</age>
      <height>64</height>
    </p:Person>
  </soap:Body>
</soap:Envelope>
```

LISTING 5 • Null references

```
Null references

package example.org.Nodes;

//Java class definition
class Node
{
String val;
Node next;

public static long ListLength (Node node );
}

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>
<next xmlns:xsi='http://www.w3.org/2001/XMLSchema'
-instance xsi:nil='1'/>
  <soap:Body >
    <n:ListLength xmlns:n='urn:example-org:nodes'>
      <node>
        <val>New York</val>
          <next>
            <val>Paris</val>
              <next>
                <val>London</val>
              </next>
          </next>
      </node>
    </n:ListLength>
  </soap:Body>
</soap:Envelope>
```

LISTING 6 • Dynamically typed data

```
package example.org.Poly;

//Java definitions
class Poly
{
public static void Execute (Object param );
```

```
}

'Visual Basic Definition
Public Sub Execute (param As Variant )
End Sub

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <p:Execute
     xmlns:p='urn:example-org:poly'
     xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
        <param xsi:type='xsd:long'>2000</param>

     </p:Execute>
   </soap:Body>
</soap:Envelope>


<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <p:Execute
     xmlns:p='urn:example-org:poly'
     xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
     xmlns:pre='urn:example-org:people'>
        <param xsi:type='pre:Person'>
          <name>
             <givenName>Martin</givenName>
             <familyName>Gudgin</familyName>
          </name>
          <age>33</age>
          <height>64</height>
        </param>
     </p:Execute>
   </soap:Body>
</soap:Envelope>
```

LISTING 7 • Simple array example

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <m:MethodResponse
       xmlns:m='urn:example-org:someuri'>
        <enc:Array
     xmlns:enc='http://schemas.xmlsoap.org/soap encoding/'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema'
enc:arrayType='xsd:long [5]'>
<enc:long>2</enc:long>
<enc:long>3</enc:long>
<enc:long>5</enc:long>
<enc:long>7</enc:long>
<enc:long>9</enc:long>
        </enc:Array>
     </m:MethodResponse>
   </soap:Body>
</soap:Envelope>
```

LISTING 8 • Multidimensional array example

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <m:Method
       xmlns:m='urn:example-org:some-uri'>
        <enc:Array
     xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'
          xmlns:xsd='http://www.w3.org/2001/XMLSchema'
          enc:arrayType='xsd:string [2,3]'>
        <item>row 1 column 1</item>
        <item>row 1 column 2</item>
        <item>row 1 column 3</item>
        <item>row 2 column 1</item>
        <item>row 2 column 2</item>
        <item>row 2 column 3</item>
        </enc:Array>
     </m:Method>
   </soap:Body>
```

LISTING 9 • Partial array transmission

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
```

```
   <soap:Body>
     <m:Method xmlns:m='urn:example-org:someuri'>
        <enc:Array
     xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema'
enc:arrayType='xsd:string [9]'
enc:offset='[2]'>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        </enc:Array>
     </m:Method>
   </soap:Body>
</soap:Envelope>
```

LISTING 10 • Sparse arrays

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <m:Method xmlns:m='urn:example-org:someuri'>
        <enc:Array
     xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema'
          enc:arrayType='xsd:string [9]'>
        <item enc:position='[1]'>Venus</item>
        <item enc:position='[3]'>Mars</item>
        <item enc:position='[7]'>Neptune</item>
        </enc:Array>
     </m:Method>
   </soap:Body>
</soap:Envelope>
```

LISTING 11 • Jagged arrays with single reference accessors

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body>
     <enc:Array
     xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:xsd='http://www.w3.org/2001/XMLSchema'
   enc:arrayType='xsd:string [][2]'>
        <enc:Array enc:arrayType='xsd:string [2]'>
          <item>Mercury</item>
          <item>Venus</item>
        </enc:Array>
        <enc:Array enc:arrayType='xsd:string [6]'>
          <item>Mars</item>
          <item>Jupiter</item>
          <item>Saturn</item>
          <item>Uranus</item>
          <item>Neptune</item>
          <item>Pluto</item>
        </enc:Array>
     </m:Method>
   </soap:Body>
</soap:Envelope>
```

LISTING 12 • Jagged arrays with multireference accessors

```
<soap:Envelope
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
soap:encodingStyle='http://schemas.xmlsoap.org/soap/
encoding/'>
   <soap:Body
     xmlns:enc='http://schemas.xmlsoap.org/soap/encoding/'
     xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
     <m:Method xmlns:m='urn:some-uri'>
        <enc:Array enc:arrayType='xsd:string [][2]'>
          <item href='#id1'/>
          <item href='#id2'/>
        </enc:Array>
     </m:Method>
     <enc:Array id='id1' enc:arrayType='xsd:string [2]'>
        <item>Mercury</item>
        <item>Venus</item>
     </enc:Array>
     <enc:Array id='id2'
          enc:arrayType='xsd:string [6]'>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
        <item>Pluto</item>
     </enc:Array>
   </soap:Body>
</soap:Envelope>
```
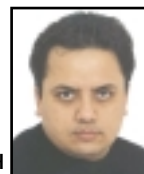
▼ Download the Code
www.sys-con.com/xml

WRITTEN BY **HITESH SETH**

XML in 2002 and Beyond

A look at the past, present, and future of XML

We're all hoping for a revolutionary year for the economy and the world of i-technology in particular. The New Year also marks an important phenomenon: all of us try to have a New Year's resolution (mine is to gain fame and riches in the world of i-technology). And of course, we tend to think of the events that marked the previous year as well as what's in store for the coming year. Here we'll look at how 2002 steered XML into the world of i-technology, and project the path of XML in 2003.

Standards

The success of XML has often been attributed to the development of standards. Whereas the W3C (World Wide Web Consortium) has taken on the responsibility of the core set of standards around XML, a number of other organizations have been formed to jump-start related horizontal- or vertical-focused standards development. OASIS (Organization for the Advancement of Structured Information Standards) has been instrumental in developing vocabularies based on XML. 2002 marked the establishment of the Web Services Interoperability Organization (WS-I), chartered to promote Web services interoperability across platforms, applications, and programming languages.

W3C

The standardization work at W3C follows a known path. W3C Working Groups take the development of a standard through a series of stages: Working Draft, Last Call Working Draft, Candidate Recommendation, Proposed Recommendation, and W3C Recommendation. Related to XML activities, 2002 marked the release of XML Encryption, XML Decryption,

XML Signature, and Exclusive XML Canonicalization as W3C Recommendations. SVG 1.1 and Mobile SVG were upgraded to Proposed Recommendation status while SOAP 1.2, Namespaces in XML 1.1, XForms, XML 1.1, and XInclude were released as Candidate Recommendations.

Apart from these Recommendations, 2002 also marked the development of key standards that are still in the Working Draft status, particularly MathML 2.0, XHTML 2.0, Voice Browser Activity (Speech Synthesis Markup Language, VoiceXML 2.0, CCXML, SVG 1.2, XQuery, XSLT/XPath 2.0, OWL, Web Services Architecture, RDF Schema, DOM Level 3, XML Accessibility Guidelines, XML Events and XFrames and WSDL 1.2).

OASIS

Key standards earning the "OASIS Standard" stamp included SAML (Security Assertion Markup Language), ebXML Messaging Service Specification, ebXML Collaboration Protocol Profile and Agreement, ebXML Registry Service, and Information Model and Directory Services Markup Language (DSML). A number of other OASIS Technical Committees (TCs) were in the news, particularly Web Services for Remote Portals (WSRP), Universal Business Language (UBL), Topic Maps, RELAX NG, LegalXML (Legal Information Exchange), Open XML Format for Office Applications, Tax XML, Web Service for Translation, and XML Common Biometric Format (XCBF).

Also, UDDI.org, which heads the development of the Universal Description, Discovery and Integration standard, transitioned and integrated its standardization activity with OASIS.

Other Initiatives

Apart from the standardization work

carried by W3C and OASIS, 2002 also saw standardization work by the SALT Form (to develop Speech Application Language Tags); Liberty Alliance released 1.1 specifications; and RELAX NG, the next-generation schema language for XML, was released as a draft ISO standard.

Beyond Web: XML to Go
XML in speech recognition

There was a great deal of interest in applying XML standards to opening up the Interactive Voice Recognition (IVR) industry. The VoiceXML Forum had already developed the initial version (VoiceXML 1.0) of an XML-based speech dialog and synthesis specification. With a memorandum of understanding, the W3C Voice Browser activity was put in charge of the development of VoiceXML 2.0. In addition to VoiceXML, Microsoft spearheaded the development of Speech Application Language Tags (SALT) through the SALT Forum organization. SALT, which shares a number of similarities with VoiceXML, is also set to promote the notion of a multimodal interface where speech can be used as one of the interfaces. The SALT Forum submitted its specification to the W3C. We also witnessed the creation of XHTML+Voice, a standard built on top of XHTML and VoiceXML to provide multimodal activity as well.

Wireless applications

We have also seen the convergence of the wireless and Web standards with the creation of WAP 2.0. By leveraging XHTML, WAP 2.0 attempts to unify the Web and wireless standards. Whether it be companies or standards, consolidation is the key. Last year saw the formation of Open Mobile Alliance, Ltd., an organization formed with the consolida-

AUTHOR BIO
Hitesh Seth, editor-in-chief of XML-Journal, is the chief technology officer of ikigo, Inc., a provider of XML and Web services monitoring and management software.

Industry/Sector	Organization	Initiative
Travel	OpenTravel Alliance	• XML Specifications Supporting Travel Industry Messaging • Supports multiple travel verticals • Supports ebXML-OTA Mapping
Biometrics	OASIS	XML Common Biometric Format
Human Resources	HR-XML Workgroup	• Employer Stock Plan Standards • Staff Industry Data Exchange
Legal	OASIS	LegalXML
Financial Services	ISDA	FpML (Financial Products Markup Language) 3.0
	European Service Providers	Securities Financing XML
	IFX (Interactive Financial Exchange)	IFX 1.3
Automotive	STAR, OAGI Covisint	XML Standards in the Automotive Retail Space Announced support for ebXML Message Specification and OAGIS Standards
Media	International Press Telecommunications	NewsML 1.1
Financial Services	NYSE/FISD	Vendor Reporting XML (VRXML)
Cross Industry	ANSI ASC X12	Reference Model for XML Electronic Business Messages
Health Care	Clinical Data Interchange Consortium	CDISC Operation Data Model
Pharmaceutical	ICH Multi-disciplinary Group 2 (M2)	Electronic Common Technical Document (eCTD)
Chemical, Petroleum and Agricultural	CIDX, PIDX and RAPID	Common platform-independent electronic business data communications standards
Retail	Association for Retail Technology Standards (ARTS)	XML Price and Digital Receipt Schemas

Table 1 • Industry initiatives

tion of multiple wireless initiatives including WAP Forum, Open Mobile Architecture, Location Interoperability Forum (LIF), Wireless Village, and SyncML.

XML for the Masses

At a recently held XML-focused conference, representatives from both Microsoft and Sun talked about using XML as the underlying format for office productivity suites. Late last year, Microsoft announced that Office 11, the latest reincarnation of its office productivity suite, would have an XML file format underneath it. We also saw the emergence of an OASIS technical committee chartered to develop an "Open Office XML File Format." The proposed XML file format is to be "suitable for office documents containing text, spreadsheets, charts, and graphical documents." Later in the year, Sun also fueled this work by contributing OpenOffice.org XML Format to the TC as well. However, whether both Microsoft and Sun will collaborate and agree on a joint file format is yet to be seen. Watch out, XML isn't just for programmers – it's going to be for the masses.

Web Services

Web services is tightly connected to XML. Without any doubt, XML has really made the entire Web services vision possible. At the same time, maybe as a payback, Web services is also emerging as probably the most recognized application of XML as well. The core Web services standards – SOAP, WSDL, and UDDI – have already been developed; however, what we saw in 2002 was sort of ownership changes. SOAP was released as a W3C Candidate Recommendation, and development of UDDI was transferred to OASIS, which has already been developing the ebXML set of open standards.

It's important to understand that even though integrating systems is really what the Web services architectures have been utilized for today, applications of Web services aren't limited to application-to-application integration. This was further highlighted by emergence of two OASIS technical committees in this direction: Web Services for Remote Portals (WSRP) and Web Service User Interface (WSUI).

In 2002, a major amount of development around Web services involved modeling and executing business processes using a standard that orchestrates a set of related Web services together to form a business process. A whole new alphabet soup was spilled in this arena, with the emergence a of number of initiatives related to this activity. Key to those are BPEL4WS (Business Process Execution Language for Web Services), Web Service Choreography Interface (WSCI), BMPL (Business Process Markup Language), and so on. One standard that seems to have a lot of activity is BPEL4WS, which has been jointly developed by IBM, Microsoft, and BEA.

In addition to BPEL4WS, the trio (IBM, Microsoft, and BEA) also collaborated together to spearhead development on related Web services initiatives, WS-Coordination and WS-Transaction. Later in the year, IBM and Microsoft also announced development of another set of standardization initiatives related to Web services security. WS-SecurityPolicy, WS-Trust, WS-SecureConversation, WS-Policy, WS-PolicyAttachment, and WS-PolicyAssertions were developed in collaboration with BEA, RSA, and SAP.

Industry Applications

Since the conception of XML, a number of industry organizations have come forward and started applying XML in their own industries. Last year wasn't different in this regard. A number of industries, including retail, travel, financial services and banking, health care, chemical, petroleum, agricultural, pharmaceutical, legal, biometrics, press, human resources, and automotive, either announced vocabulary standardization initiatives or built on top of existing work. Table 1 highlights some of the initiatives of various industries.

Product Initiatives

The last year saw a strong lineup of products related to XML. No doubt support for XML in Microsoft Office and OpenOffice.org topped the news. The "XML for Masses" philosophy was adopted by Adobe, which announced support of XML in its print media products, an area where I see a lot of possible growth.

(Adobe announced XML export/import capability in its FrameMaker and InDesign products.) Also, we saw a rise in native XML storage products – whether it was existing relational or OODBMS vendors or new niche vendors claiming optimized XML storage and indexing.

2002 Favorites

Looking back, my favorite developments in the XML world in 2002 were:

- XML for the Masses Initiative: XML file format in Office productivity suites.
- ebXMLseems to near its vision of being a true electronic business XML standard.
- XML security-related efforts: XML Signature, XML Encryption/Decryption (Security Assertion Markup Language [SAML]).
- Convergence of wired/wireless Web, with XHTML being the key.
- XML for speech/multimodal interactions: VoiceXML, CCXML, SALT, XHTML+Voice
- XML Schema is being adopted as the basis for a number of different vocabularies. (Of course, we have RELAX NG as a true alternative.)

- XML 1.1 with Unicode support.
- Formation of the Web Services Interoperability Organization.
- A number of initiatives focused around Web services choreography/ orchestration/process modeling initiatives.
- XML Web services emerge as the single most important and achievable mechanism for J2EE/.NET interoperability.
- Availability of Amazon, Google Web services, which made the Web services vision really come out of the typical "enterprise integration" area and applies Web services for interesting applications for the Web itself.
- A number of initiatives triggered to utilize XML in government across the world.

Looking Forward: XML in 2003

Enough of the past, let's look forward. So what's in store for 2003? Here are my predications, what I look forward to for XML in 2003:

- Official W3C stamp on Web services core standards (SOAP, WSDL, etc.).
- XQuery, XPath.XSLT 2.0.

- Some sort of consensus on the RELAX NG and/or XML Schema debate.
- Progress around the recently released XForms initiative.
- Official W3C stamp on speech/multimodal dialog language (SALT and/or VoiceXML).
- Once the core Web services standards are in place, a lot of activity around Web services transaction management, security, management, etc.
- A lot of activity and some key deliverables from the Universal Business Language (UBL) OASIS TC.
- Web Services User Interface and integration of Web services with enterprise portal frameworks.
- Web services management.
- An industry consensus toward a common Web services choreography/ orchestration standard.
- Gradual ebXML adoption within enterprises.
- XML to be more widely adapted in media (integration with publishing systems), acceptance/developments in standards such as XSL-FO. ⬙

HITESH@SYS-CON.COM

XML INITIATIVES
XML in e-Government
Changing how government works

PRODUCT REVIEW
Microsoft BizTalk Server 2002
Enabling integration and automation

THE SEMANTIC WEB
Ontologies for Knowledge Representation
A vision of the future

WEB SERVICES
Building a Real-World Web Service - Part 2
Process choreography and orchestration

DON'T MISS XML-J MARCH

XML FOR INTEGRATION

Sitraka
(now part of Quest Software)
www.sitraka.com/jclass/ws

Altova

http://xmlj.altova.com/xmlspy5