Content Management • Data Management • XML Labs

XML JOURNAL

THE ULTIMATE XML ENTERPRISE RESOURCE

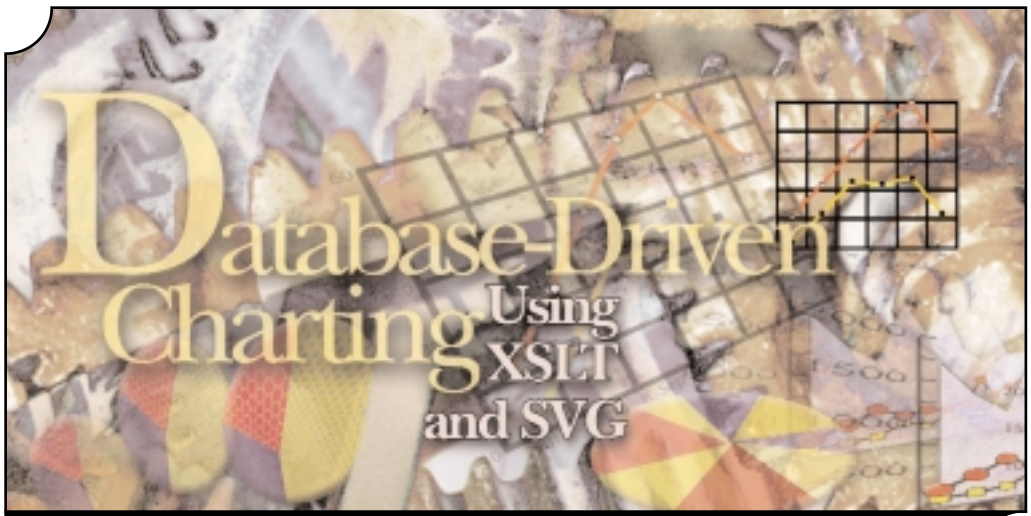
May 2003 Volume: 4 Issue: 5

xml-journal.com

XML Edge 2003 East
International XML Conference & Expo
SPECIAL SHOW REPORT

$6.99US $7.99CAN
06
0 09281 01314 3

SYS-CON MEDIA



Controlling finer details of data-driven Scalable Vector Graphics

XML JOURNAL

FOUNDING EDITOR
Ajit Sagar ajit@sys-con.com

EDITORIAL ADVISORY BOARD
Graham Glass graham@themindelectric.com
Coco Jaenicke cjaenicke@attbi.com
Sean McGrath sean.mcgrath@propylon.com
Simeon Simeonov talktosim@polarisventures.com

EDITORIAL
Editor-in-Chief
Hitesh Seth hitesh@sys-con.com
Editorial Director
Jeremy Geelan jeremy@sys-con.com
Managing Editor
Jennifer Van Winckel jennifer@sys-con.com
Editor
Nancy Valentine nancy@sys-con.com
Associate Editors
John Evdemon jevdemon@sys-con.com
Jamie Matusow jamie@sys-con.com
Gail Schultz gail@sys-con.com
Jean Cassidy jean@sys-con.com

PRODUCTION
Production Consultant
Jim Morgan jim@sys-con.com
Art Director
Alex Botero alex@sys-con.com
Associate Art Directors
Louis F. Cuffari louis@sys-con.com
Richard Silverberg richards@sys-con.com
Assistant Art Director
Tami Beatty tami@sys-con.com

CONTRIBUTORS TO THIS ISSUE
Joel Amoussou, Michael Brauer, Bing Li, Douglas Lovell, Glen Martin, Suhayl Masud, Avinash Moharil, Patti Purcell, Hitesh Seth, Carl Sjogreen, Wei-Tek Tsai, Rajesh Zade

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9637
XML-JOURNAL (ISSN# 1534-9780)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices.
POSTMASTER: Send address changes to:
XML-JOURNAL, SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

SYS-CON MEDIA

The New Role of XML

WRITTEN BY **CARL SJOGREEN**

In the history of XML to date, its role in application development has been mostly on the edge – it has been used primarily as the format for applications to communicate with each other, as a way to serialize data or configuration information, or for some other use at the "front door" of the application. The internal data model and processing that made applications run were entirely driven by objects (Java, C#, or what have you), relational database schema, and the like. Developers used the same approach to data modeling they always had and leveraged XML on the outside of their applications.

As XML has become more mainstream, and there is much more XML data floating around the enterprise, new technologies have been developed to better process and manipulate XML inside applications. As a result, the role and location of XML use in applications is shifting. We are starting to see more and more applications in which XML is not just a way to serialize or communicate the already extant data structures, but a way to think about modeling the application data itself.

To get concrete for a moment, think of a Web services application for handling an order. The Web service might define "Customer" data and a set of "LineItem" data that makes up the order. Usually great care goes into defining the structure and the associated validation rules of this data, because it's a lot easier to enforce these constraints in a declarative medium like XML Schema than in a programming language. Today a Web service like this is usually just a thin "XML" layer fronting a legacy application. XML is being used at the edge, and is translated into the internal application data objects (Customer and Order for example) where the real work happens.

New order processing applications, however, are being built with Web services in mind from the get-go, not as a layer tacked on after the fact. In these applications, where you aren't constrained by many existing data structures, you start to wonder why you need to translate this nice data model you defined in XML Schema into the less expressive type systems of a programming language just to write your application logic, only to translate it out to XML again. Moreover, you have to write a lot of code to make sure that the data continues to meet the original constraints imposed by the schema as it flows through the system.

Part of the reason this hasn't been done to date is that accessing XML from within programming languages is just too hard. Either you are forced to deal with low-level XML processing like DOM or SAX, or you have to live with the loss of schema information that comes from translating XML into programming language objects. New technologies like BEA XMLBeans (http://dev2dev.bea.com/technologies/xmlbeans/index.jsp) are starting to change the productivity trade-off that developers are normally forced to make. XMLBeans allow you to easily access XML data directly using XQuery and an XML cursor, but also automatically build Java "views" on your XML data based on information provided by XML Schema. This allows you to have the convenience of accessing data using a Java programming model, but with all the constraints of the data model defined by XML Schema still in force, and the inherent extensibility of XML always available to you. Similarly, the ECMA group has been working to make XML a first-class citizen in the JavaScript programming language (www.ecma-international.org/news/ECMA%20E4X%20Final%20Final.pdf), and rumors abound that Microsoft has a similar initiative in the works with X#.

With tools like these, developers can start to think seriously about using XML Schema and XML as the center of the data model of an application, not just as a translation that happens at the outside. To be clear, it's always important to have some translation layer at the edge of your application to ensure loose coupling between the internal data model and the public data, but there is no reason this has to be a translation between XML and objects. If XML is used as the internal data model, it can be a simple XML-XML transformation, for which there are many tools and languages available.

So in the next application you write, consider using XML at the center. If you are building an application that does a lot of XML processing, I think you'll find that using some of the new technologies for accessing XML gives you the same productivity you'd expect, and saves you from the tedious translation and validation code you're normally forced to write. I think we'll see XML permeating application development more and more over the coming years, and a whole set of new tools and technologies being developed to unite relational, object-oriented, and XML data even more over time.

Author Bio

Carl Sjogreen is senior product manager for BEA WebLogic Workshop, an integrated development framework that makes it easy for all developers - not just J2EE experts - to build enterprise-class Web services on the WebLogic platform. Carl has been involved with XML, Web services, and developer tools since 1998, when he founded Transformis, developers of the award-winning Stylus Studio IDE.

CARLS@BEA.COM

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9637

Who Owns Your Data?

WRITTEN BY **MICHAEL BRAUER**

The incompatibility of today's proprietary file formats goes well beyond the inconvenience of, say, unreadable e-mail attachments. It raises the larger issue of ownership – and cost of ownership.

The data in your spreadsheet, the content in your business presentation, the words in your word processor – all of these belong to you. You created them. But today most of these documents are stored within binary formats, which means they're worthless without the applications that created them. (That's like owning a car but having to ask someone else for the keys every time you want to drive it.) Worse, there's no guarantee, given today's undocumented, proprietary formats, that these documents will be readable even five years from now. This makes archiving complicated and costly.

In an increasingly connected world, you should also be able to share the content you create with anyone you choose, whether or not they have the same software you use. And you should be able to process that content with applications that have a completely different purpose (workflow, content management, etc.) without the burden and expense of dealing with multiple document formats.

Our goal is to achieve consensus on an open standard that will protect content, whether it's an 800-page airplane specification or a legal contract, from being locked into a proprietary file format, while simultaneously opening new possibilities.

By "our goal" I mean that of a recently formed technical committee for an international standards body known as OASIS, the Organization for the Advancement of Structured Information Standards.

We are working to define an XML Schema that is suitable to represent the structure of office documents – a schema that can be used to process and archive the millions of text documents, spreadsheets, and presentations that have been created and will be created. There is currently no standardized schema for such documents, but there is a strong demand for it.

The benefits are clear:

- A standard method for processing and interchanging office documents will enable companies to own their data and freely choose tools to view and edit that data long after the originating applications have come and gone. (In fact, a well-documented standard should mean that documents will be readable even 50 years from now.)
- Having a standard format for office productivity applications also makes it much simpler to create other kinds of applications that work on top of them, from simple search engines to complex information management systems.

The alternative approach – having businesses write their own XML Schemas – would simply add unnecessary overhead with no guaranteed cost benefits. It seems rather old-fashioned to assume that XML tagging can be made painless for the individual or that every enterprise will want to write its own schemas. Over the past decade, experience has shown quite the opposite. In a user-defined schema, serious overhead will always be added because (by definition) it adds human judgments and human errors that you wouldn't get by turning some smart software loose on the document instead.

> **"...open-standard file formats will enable us to concentrate on what's really valuable about any document – the content, not the program used to create it"**

Experience has also shown that there's generally no big ROI in designing a custom schema for every installation. The ROI is in designing a few big schemas that express semantic agreements across an entire industry in a way that allows for small variations to fit individual data exchange relationships. In our view, standardization offers a much better investment for the industry overall as well as for individual businesses.

Office suites, already used in a wide range of industries, clearly meet basic business needs, which makes them an ideal place to establish a basic XML Schema that many companies can share. This would allow companies to start using XML, which many recognize as advantageous, without having to complete a lengthy analysis of all their industry-specific requirements.

Most important, open-standard file formats will enable us to concentrate on what's really valuable about any document – the content, not the program used to create it.

Author Bio

Michael Brauer, of Sun Microsystems, is chairman of the OASIS Open Office XML-Format Technical Committee.

MICHAEL.BRAUER@SUN.COM

Building a Real-World Web Service – Part 4

The extended e-business scenario

WRITTEN BY
SUHAYL MASUD

It's time for Web services to prove their worth. They've gained enough attention for people to stop and notice, to wait and expect, to speculate and argue…now it's time to show people what the fuss is all about.

If the show fails to impress, Web services will slowly fizzle out like their predecessors. To make Web services succeed, the Web services community needs to choose winners from the competing Web services standards, vendors need to present a coherent message, and the community needs to keep stressing the true value of Web services: enabling e-business dialogue between business partners. This series of articles is focused on creating robust e-business dialogues using a mix of Web services standards and RosettaNet.

In Part 3 of the series, I showed you the components and tools necessary to construct and conduct the e-business dialogue, and how to install them. I constructed the public side of the e-business dialogue by building 6 of the 13 components necessary for the extended e-business scenario. In this article, I will show you how to build the private side of the process, thus completing the implementation of the extended e-business scenario. In this article I will also discuss the challenges of creating real e-business dialogues with Web services.

The Extended E-Business Scenario

The extended e-business scenario that we are building in this series is illustrated in Figure 1. To recap, the extended e-business scenario works as follows: after a particularly successful day of selling laptops, the Inventory Manager application at ACME realizes the inventories are running low, and it triggers the ACME PO requester process. The requester process then begins a public e-business dialogue with Laptops, Inc., placing a purchase order request.

Laptops, Inc., receives the purchase order request from ACME and immediately sends back a message acknowledging receipt of the request. The PO requester process at ACME receives the receipt acknowledgment and informs the Inventory Manager that the purchase order has been placed. The Inventory Manager returns to other activities and the PO requester ends the request process. Meanwhile at Laptops, Inc., the place PO process invokes the Sales Agent, an internal Web service, and passes ACME's purchase order request to the Sales Agent. The Sales Agent figures out which parts of the purchase order can be fulfilled and forms a purchase order confirmation that provides a detailed line-by-line account of the items it can provide. This confirmation is then passed from the Sales Agent back to the public place PO process at Laptops.

The process at Laptops then sends the purchase order confirmation to an ACME process that accepts purchase order confirmations. The ACME confirmation acceptance process invokes an internal Web service to inform the Accounts Service at ACME about the purchase. The ACME process next sends back a message acknowledging the receipt of the purchase order confirmation and concludes the purchase order e-business dialogue between ACME and Laptops, Inc. Laptops, Inc., receives this receipt acknowledgement and invokes an internal Web service to its shipping department so that the order can be shipped to ACME.

As you can see, there is a lot of activity taking place in this scenario. Before we get into the implementation end, it is important to explore how to create realistic processes, and areas where Web services have room for improvement.

Building Reliable Asynchronous Processes

To enable a realistic public process between the two business partners, it is necessary to create it as an asynchronous process, the key to flexible and longer running e-business dialogues. Remember that BPEL processes are layered over Web services, adding business processes to the basic functionality provided by Web services. BPEL processes introduce stateful interactions, co-relate independent Web services, and define partners, activities, data containers, condition statements, and many other features to provide flexible and more descriptive e-business dialogues.

Web services perform synchronous operations by default. This means that a service requester sends a request and waits until a response is received from the service provider.

To create a synchronous purchase order process where ACME sends a PO request to Laptops, Inc., the only component required would be to build a PO Web service at Laptops, Inc. This service would take the request from ACME, process it, and send a response on the same connection, within a matter of seconds. What happens if the connection is lost? What if it takes longer to furnish a response? While ACME is waiting,

how will it know if Laptops is busy working on its requests, or if Laptops never received the request? This method of communication has two problems: it is synchronous and unreliable.

The "business transactions" are typically longer running activities that might take hours to respond, and in these transactions, both parties need to rest assured that the messages are being received. A simple way to enable reliable messaging is for each party to send back a simple "signal" acknowledging receipt of the message. In this mode, even if Laptops, Inc., takes two hours to reply to the PO request, it will acknowledge the receipt of the PO request immediately. This way ACME knows that the request order has been received; otherwise ACME might start sending the same PO request repeatedly, causing confusion and hindering the e-business dialogue.

To enable asynchronous communication using Web services, Laptops, Inc., needs to implement a Web service but so does ACME. When ACME sends a purchase order request, it receives only a receipt acknowledging that request. Next, Laptops, Inc., would invoke a Web service on ACME and send the purchase order confirmation as a request, and receive a receipt acknowledging the purchase order confirmation as the response. For the real-world Web service, I have extended this simple idea into BPEL processes, to allow for stateful, asynchronous, reliable interactions between business partners.

Room to Improve in BPEL4WS and WSDL

The discussion of building real-world Web services would not be complete if I didn't point out that WSDL and BPEL4WS are both young efforts, with some inadequacies that need to be addressed in "version 2" releases. The encouraging news is that the W3C has started several groups to look at various aspects of Web services including the Web Services Architecture Group, the Web Services Description Group, and more recently, the WS Choreography group. These groups include veterans from EDI, RosettaNet, and ebXML, who will hopefully keep the groups from reinventing the wheel, and leverage solutions from the more mature standards.

If we look strictly at the WSDL side, the Web services definition encourages a relationship that is more of a service provider serving an anonymous service requester. A WSDL file simply tells a client how to access the services; it has no provisions for the service provider to find out about the service requester. A real e-business dialogue requires a more peer-to-peer role, meaning that both partners can be providers and requesters. In WSDL, the service provider cannot initiate the dialogue; it has to wait for a request and can only send a response. This setup makes it difficult to establish colleague-to-colleague relationships, and awkward to construct dialogues that require asynchronous operations.

BPEL4WS is a very recent effort and will take time to mature. One of the problems of working with BPEL was using an even younger alpha version BPWS4J engine. The BPWS4J engine does not implement the entire BPEL4WS spec, which caused a fair share of problems. The most significant problem is that at the time of this writing, the BPWS4J engine allows only RPC calls. Sending the document as an RPC call causes problems when you have to pass larger documents to your BPEL process, as it is a tedious process.

Since BPEL4WS is layered on top of Web services, a simple e-business dialogue becomes quite complicated to build. The public process between business partners should really take two public interface definitions; instead, it takes three BPEL processes and two Web service definitions.

Building the E-Business Dialogue

Now that we've discussed how to build reliable asynchronous processes, and the challenges we face building them, it's

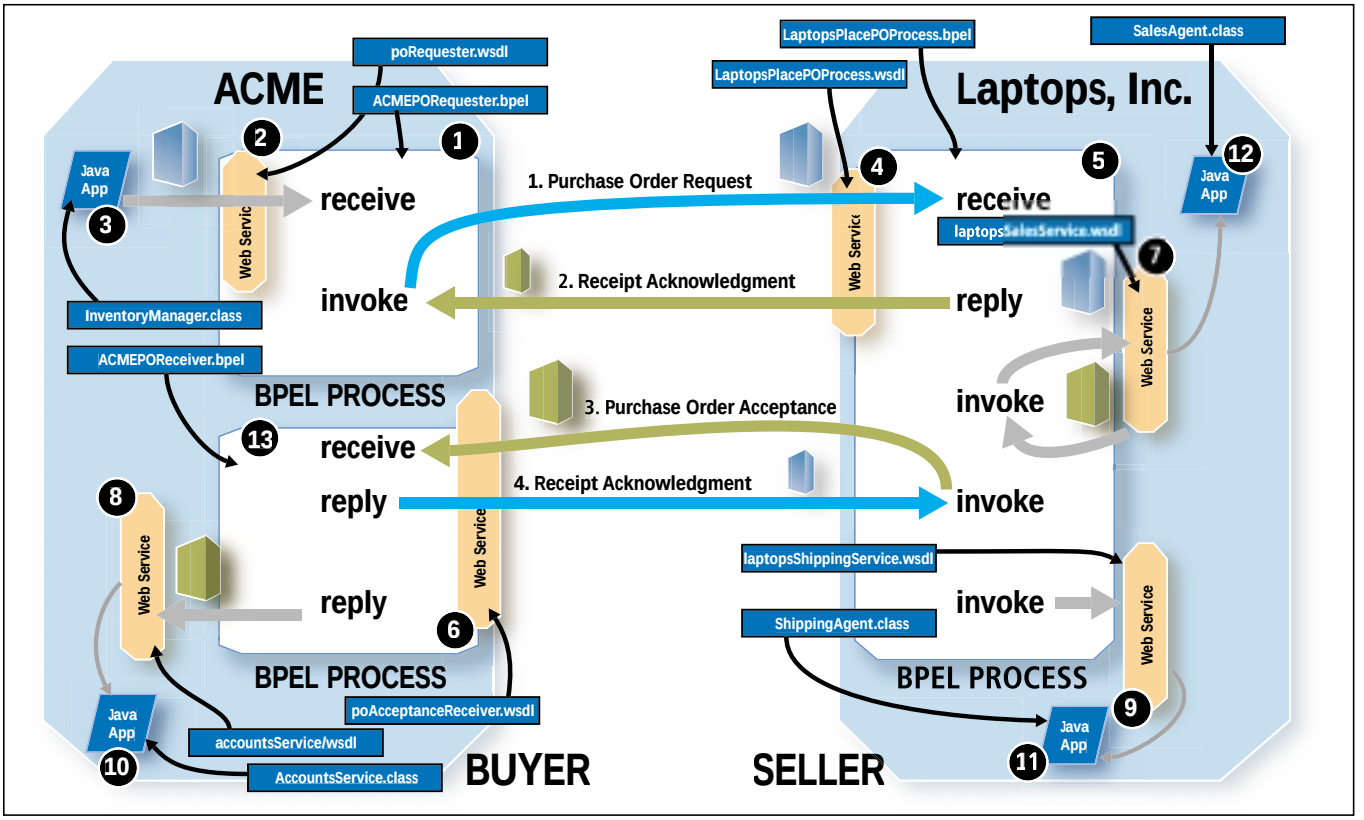


Figure 1 • The extended e-business scenario using BPEL, WSDL, and Java components

time to finish the construction process we began in Part 3 of this series. But first we need to acquaint ourselves with the languages, tools, and components involved in the building process.

We need a variety of languages and tools to build and execute the e-business dialogue. We are using the definition of an e-business dialogue, the choreography and messages definitions, from RosettaNet. The e-business dialogue is written using a mix of WSDL and BPEL4WS, which are both XML-based languages, and the internal Web services interact with applications written in Java.

The tools needed for creating and conducting the scenario are Eclipse, BPEL Editor, Tomcat, and the BPWS4J engine. Eclipse is needed to write and compile Java classes. It's also needed for the BPEL Editor plug-in to build and compile the BPEL process. We need the BPWS4J engine, which runs on Tomcat, to execute a business process. To get in-depth descriptions of these tools and how to install them, please refer to Part 3 of the series.

The Components

Thirteen components are needed to conduct this end-to-end business scenario (see Table 1). The public part of the e-business dialogue only requires five components, but since that may leave readers wondering how to tie the back-end systems to the public process, I am building the extended e-business scenario, complete with the back-end hooks.

Inventory Manager

This Java application resides on the enterprise server at ACME. It sends an internal purchase order request to the ACME PO requester process using SOAP calls. The Inventory Manager receives a receipt acknowledgment verifying that the purchase order has been placed.

It is important to understand how Apache SOAP works to understand how Java applications can interact with Web services. The Apache SOAP user guide is available online at http://ws.apache.org/soap/docs/index.html. Listing 1 shows how Inventory Manager makes the call.

The Inventory Manager makes a SOAP call to the PO Requester Web service. The call object needs to know the URI where the call will be invoked; this URI is provided by the BPWS4J engine after the ACME PO requester process is installed. Next, the method that the call will invoke is set. This method name matches the operation name of the PO Requester Web service. The Inventory Manager performs an RPC-style SOAP call; therefore, the call object needs to know the parameters passed to the method. Note that the name of the parameter, PORequest, matches the part name of the input message on the PO Requester Web service.

PO Requester Web service

This internal Web service, written in WSDL, resides on the ACME server and is consumed by the ACME PO requester process. The Inventory Manager sends its purchase order request to this Web service, and since the process is layered on top of this Web service, it takes care of the request. In our example, the requester Web service returns a receipt acknowledgment from Laptops, Inc., that the purchase order has been received. The complete source code file is available for download from www.sys-con.com/xml/sourcec.cfm; Listing 2 is the snippet we are interested in.

PO Receiver BPEL process

This BPEL process is a public e-business dialogue component at ACME. Its job is to receive the purchase order acceptance sent by Laptops, Inc., send back a receipt acknowledgment to Laptops, Inc., and invoke the Accounts Service internally at ACME. The Accounts Service is a Java application, but since BPEL processes can interact only with Web services, we need to define a Web service "wrapper" for the Java application. Let's start by making the necessary additions to the process. First, we need to add the Accounts Service as a partner:

```
<partners>
      … <partner name="Accounting"
serviceLinkType="acmeAcct:AccountsServiceSLT"/>
  </partners>
```

Next, we need a container definition, see PurchaseReport container below:

```
<containers>
```

ID	Component name	Language	File Name	Partner
1	ACME PO Requester Process	BPEL4WS	ACMEPORequester.bpel	ACME
2	PO Requester Web Service	WSDL	poRequester.wsdl	ACME
3	Inventory Manager	Java	InventoryManager.class	ACME
4	Laptops Inc Place PO Web Service	WSDL	LaptopsIncPlacePO.wsdl	LAPTOPS Inc
5	Laptops Inc Place PO Process	BPEL4WS	LaptopsPlacePOProcess.bpel	LAPTOPS Inc
6	PO Acceptance Receiver	WSDL	poAcceptanceReceiver.wsdl	ACME
7	Laptops Sales Service	WSDL	laptopsSalesService.wsdl	LAPTOPS Inc
8	Accounts Service Web Service	WSDL	accountsService.wsdl	ACME
9	Laptops Shipping Service	WSDL	laptopsShippingService.wsdl	LAPTOPS Inc
10	Accounts Service Application	JAVA	AccountsService.class	ACME
11	Shipping Agent Application	JAVA	ShippingAgent.class	LAPTOPS Inc
12	Sales Agent Application	JAVA	SalesAgent.class	LAPTOPS Inc
13	ACME PO Receiver Process	BPEL4WS	ACMEPOReceiver.bpel	ACME

Table 1 • The components required for the end-to-end e-business scenario

```
    … <container name="PurchaseReport"
messageType="acmeAcct:reportPurchaseMsg"/>
</containers>
```

The process invokes the service using a normal invoke activity:

```
<invoke name="invoke" partner="Accounting"
portType="acmeAcct:AccountsServicePT"
            operation="reportPurchase"
inputContainer="PurchaseReport" outputContainer=
  "POAcceptanceReceipt"/>
```

To see how this invoke will result in the activation of the Java application, let's look at the Accounts Service Web service.

Accounts Service Web service

This Web service, written in WSDL, is an internal Web service that enables the process to invoke a Java application (see Listing 3).

It's important to note that the parameter of the method call in the Java application matches part name "purchaseReport" of the input message "reportPurchaseMsg". Having defined the messages and the portType, the bindings section will map the Web service to the Java application (see Listing 4).

Listing 4 shows that the binding type is "JavaBinding". The binding also contains a typemap, and in our case it is simply mapping the xsd:string to java.lang.String. The next lines show that the operation reportPurchase in the Web service definition maps to the reportPurchase method in the Java class. The service element identifies the port and bindings as Java, and provides the address of the class. To make the class accessible, you can add it to a WAR (Web archive) file, and deploy it on Tomcat, or an easier method is to simply place the class file on the "\Apache Group\Tomcat 4.1\webapps\bpws4j\WEB-INF\classes" directory.

Now let's look at the Java class this Web service is bound to.

Accounts Service application

The Accounts Service application is used to report purchases made by ACME. It has the single method reportPurchase that returns a String and takes a String named purchaseReport as a parameter. The parameter name is identical to the input message part name in the accounts service Web service definition.

The AccountsService simply writes out the purchase report and a time stamp to a file named "PurchaseReport.acme" (see Listing 5).

Let's look at another example, this one defined at the Laptops, Inc., side. At the end of the Laptops place PO process, an internal Web service is invoked with a ship order. To enable this the partner and container definitions are created and then the Web service is invoked:

```
    <partner name="internalShipping"
serviceLinkType="tns:InternalPOShippingSLT"/>
…
    <container name="internalShippingOrder" mes-
sageType="tns3:shipPurchaseOrder"/>
…
   <invoke name="invokeShipping" partner="internalShip-
ping" portType="tns3:ShippingServicePT"
            operation="shipPO" inputContainer="internal-
ShippingOrder" outputContainer="internalSOResp"/>
```

The above is just the snippet of code I want to describe here; I encourage you to download the entire source for this article. Looking back at the definition of the invoke activity, it is important to note that the input container is the message sent as a "request" to the service being invoked, and the output container is the message being sent as the "response" from the invoked service. The input and output containers should contain the same "types" of parts as those expected in the input and output messages of the port type being invoked in the target Web service. Let us now look at the Laptops Shipping Service that the process interacts with.

Laptops Shipping Service Web service

The Web service shown in Listing 6 has an operation named "shipPO" that takes a message with two parts: the purchase order acceptance, and the receipt acknowledgment that ACME has received the purchase order acceptance. Listing 7 shows the binding information.

Again, we see that the Web service is using Java bindings, the only mapping required is to show xsd:string maps to java.lang.String. The Web service operation name is shipPO, which is also the method name of the Java class. In the service definition section, we see that the class is ShippingAgent. To make it work, we need to drop it in "\Apache Group\Tomcat 4.1\webapps\bpws4j\WEB-INF\classes". Listing 8 shows the definition of the Shipping Agent.

The Shipping Agent is a simple Java application that takes a confirmed purchase order and its receipt acknowledgment and writes them into a class named ShipOrders.laptops.

Using similar methods to define Java bindings, we can add the remaining components to the Laptops place PO process. Please look at the full source code to see the definitions of all 13 components used in this article.

Running the Scenario and Checking the Results

I've designed this scenario to be deployed on separate machines; to deploy it on a single machine, you'll need to run two instances of Tomcat (instructions are available in Tomcat's read-me file). To enable the Java applications for the processes, you will need to place the class files on the Tomcat server at: "\Apache Group\Tomcat 4.1\webapps\bpws4j\WEB-INF\classes". Next, launch Tomcat and deploy the processes following the instructions given in Part 3.

The scenario will create three files: ShipOrder.laptops, PORequest.laptops, and PurchaseReport.acme. The Inventory Manager and the Sales Manager come with hard-coded purchase order requests and purchase order acceptance based on RosettaNet purchase order processes. To check the results, simply inspect the three files created by the scenario.

A Summary of Building a Real-World Web Service

I started this series to prove that Web services can be used even today to conduct real e-business dialogues. The series covered important e-business architecture issues and constructed the real-world Web service piece by piece. In Part 1 I explained e-business dialogues, RosettaNet, and WSDL, and I demonstrated how to build a Web service definition using components from RosettaNet.

In Part 2 I talked about choreography and e-business standards like RosettaNet and ebXML, which provide choreography and other advanced e-business dialogue functions. I also explained how BPEL works and ended the article by creating an abstract business process that could be used as a collaboration agreement between business partners.

In Part 3 I explained how to create executable asynchronous business processes and how to implement these processes, providing the list of tools to be installed and instructions for installations. I ended the article by demonstrating the purchase order process running between two business partners.

In this installment, I have shown you how to hook your BPEL processes to your back-end Java applications. I also explained some of the challenges of building the real-world Web service. We now have a running e-business dialogue scenario that is based on RosettaNet concepts and components, described in WSDL and BPEL, built using open source tools, and deployed on an open source application server.

Are we there yet? Almost. In the real world, we cannot exchange this sensitive data on open networks without adequate protection. The next installment in this series is all about providing a secure real-world Web service. So, are Web services secure? Tune in next month to find out the answer.

Author Bio

Suhayl Masud is the founder and lead consultant at Different Thinking, a consulting firm that enables organizations to conduct electronic business, by providing training, architecture, and application construction services. Suhayl's experience includes consulting as the lead technical architect for RosettaNet, where he helped define the next generation of e-business process standards.

SUHAYL@DIFFERENTTHINKING.COM

LISTING 1

```
    Call call = new Call ();
    //The URI below is obtained by deploying the BPEL process and
       WSDL files on the BPEL engine
    call.setTargetObjectURI ("http://www.acme.com/services/
 poRequester#ACMEPORequesterServiceBP#inventoryService#http:
  //www.acme.com/services/poRequester#replenishRequestPort");
    call.setMethodName ("replenishRequest");
    call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
    Vector params = new Vector ();
    params.addElement (new Parameter("PORequest", String.class,
      poRequest, null));
    call.setParams (params);
    Response resp = call.invoke (/* router URL */ url, /*
     actionURI */ "" );
```

LISTING 2

```
<message name="ReplenishRequestType">
      <part name="PORequest" type="xsd:string"/>
   </message>
   <message name="ReceiptAckType">
      <part name="receiptAck" type="xsd:string"/>
   </message>
```

```
<portType name="replenishRequestPort">
      <operation name="replenishRequest">
         <input message="ACME:ReplenishRequestType"/>
         <output message="ACME:ReceiptAckType"/>
      </operation>
   </portType>
```

LISTING 3

```
<message name="reportPurchaseMsg">
    <part name="purchaseReport" type="xsd:string"/>
  </message>
  <message name="receiptAck">
    <part name="receipt" type="xsd:string"/>
  </message>
  <portType name="AccountsServicePT">
    <operation name="reportPurchase">
      <input message="tns:reportPurchaseMsg"/>
      <output message="tns:receiptAck"/>
    </operation>
  </portType>
```

LISTING 4

```
<binding name="JavaBinding" type="tns:AccountsServicePT">
    <java:binding/>
    <format:typemapping encoding="Java" style="Java">
      <format:typemap typename="xsd:string"
formattype="java.lang.String" />
    </format:typemapping>
    <operation name="reportPurchase">
      <java:operation methodName="reportPurchase"/>
      <input/>
      <output/>
    </operation>
  </binding>

  <service name="AccountsService">
    <documentation>Accounts Service</documentation>
    <port name="JavaPort" binding="tns:JavaBinding">
      <java:address className="AccountsService"/>
    </port>
  </service>
```

LISTING 5

```
public class AccountsService {
  public String reportPurchase(String purchaseReport) {
try{ FileWriter fwriter = new FileWriter("PurchaseReport.acme");
fwriter.write(purchaseReport + " TIMESTAMP: " + new
String(Long.toString(System.currentTimeMillis())));
  fwriter.close();
  return "Accounts Department Received Purchase Report";}
 catch(IOException exception){
System.out.println("AcmeResultWriter has encountered an exception "
  +exception.getMessage());
  return "Please resend the Purchase Report";  }}}
```

LISTING 6

```
<message name="shipPurchaseOrder">
   <part name="shipPOAcceptance" type="xsd:string"/>
   <part name="poShippingOrder" type="xsd:string"/>
</message>
<message name="shipPOResp">
    <part name="shipResponse" type="xsd:string"/>
</message>
  <portType name="ShippingServicePT">
    <operation name="shipPO">
      <input message="tns:shipPurchaseOrder"/>
 <output message="tns:shipPOResp"/>
    </operation>
  </portType>
```

LISTING 7

```
  <binding name="JavaBinding" type="tns:ShippingServicePT">
    <java:binding/>
    <format:typemapping encoding="Java" style="Java">
      <format:typemap typename="xsd:string"
formattype="java.lang.String" />
    </format:typemapping>
    <operation name="shipPO">
      <java:operation methodName="shipPO"/>
      <input/>
      <output/>
    </operation>
  </binding>

  <service name="ShippingService">
    <documentation>Laptops Inc Shipping Service</documentation>
    <port name="JavaPort" binding="tns:JavaBinding">
      <java:address className="ShippingAgent"/>
    </port>
  </service>
```

LISTING 8

```
public class ShippingAgent {
public String shipPO(String shipPOAcceptance, String
  poShippingOrder)
{String timeString = new String(Long.toString(System.
  currentTimeMillis()));
try{FileWriter fwriter = new FileWriter("ShipOrder.laptops");
fwriter.write("Timestamp: "+timeString + "\n\nThe confirmed
  purchase order: " + poShippingOrder);
 fwriter.write("\n\nThe acceptance confirmation: " +
  shipPOAcceptance);
 fwriter.close();
 return "null";}
 catch(IOException exception){
System.out.println("Shipping Agent at Laptops Inc. has encountered
  an exception " +exception.getMessage());
  return "hello";}}}
```

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

WRITTEN BY **DOUGLAS LOVELL**

Statements on Demand Using XSL-FO

Online presentation that offers better quality

We encounter many statements and reports in our day-to-day lives. Phone bills, utility bills, bank statements, and investment account statements are a few examples. More and more, the companies that produce these statements are making them available online, on demand through the Web. Some companies now offer incentives for customers to accept online statements in lieu of printed statements delivered through the mail; customers who want a printed record must print it locally.

The problem with printing locally is that statements produced as HTML don't print well. The page breaks don't occur in the right place – table footers appear at the top of the page following where they should appear, and lines of text split from one page to the next. The edges of the reports get truncated on the printer such that all of the account detail on the right-hand side is lost. A much better format for presenting statements online, from the perspective of print, is PDF, Adobe's document format that has been widely adopted for typeset-quality presentation on the Web.

Implementations of the XSL standards make it relatively simple to produce online account statements on demand as PDFs, with quality equaling that of statements that are printed and sent via mail. The benefit to vendors is that they may produce statements using the same technology for print and for online delivery. IBM now sells a product for producing AFP, a print format that drives the high-speed printers typically used to print statements for mass mailing, from FO. Using a single transform for both online and print presentation reduces development and maintenance costs and ensures consistency. Customers benefit by receiving PDF documents that they can store for later review or print with quality approaching that of statements they're accustomed to receiving in the mail.

This article demonstrates the capability for generating statements online as PDF by implementing a real-world example. The example statement is an investor account summary from an IRA. As they used to say on "Perry Mason," the names have been changed to protect the innocent.

The first stage of implementation is to write a back-end routine that accepts the customer query and produces an XML-encoded representation of the result. This is at least half of the work, but it's the same work any report generator producing the final formatted report would have to do. The advantage of producing an XML document as the end product of this step is that now all the logic for styling and presenting the content is separate from the logic needed to fetch it. A change to the report format will not necessarily entail digging into code that both fetches and formats the data. XML decouples data fetching from data formatting.

One major decision that arises when preparing the XML that will drive a report is whether to include summary and calculated information. Including summary information moves computation to the back-end process that makes the database queries and produces the XML. Excluding summary information moves computation to the transform process that styles and presents the data. Some forms of computation may be easier or more efficient when done by the SQL engine or programming logic than when done with the transform. The transform engine may do simple subtotals, but computations such as market value, yield, or asset percentages might best be done with the back-end logic.

One way to tackle a formatting task like this one is to use a top-down approach. Start with the overall layout and structure of the document and work down into the details of each section. The main challenge is to ensure that the content of each section ends up inside an fo:block, where text is allowed. Listing 1 provides the first coarse cut. It provides the page layout, section headers, and block elements for the text. Account information appears in a static header region that will repeat at the top of each page. The body of the statement contains transaction and summary information. Figure 1 illustrates the first intermediate result. It doesn't look particularly well formatted at this point, but it runs. It's a start – it's something that works, that you can refine.

You can apply two forms of refinement when incrementally developing a stylesheet: structural refinements and refinements of style. Structure refers to the basic elements and their nesting, one within another. Style refers to the properties that control visual elements such as color, font, borders, and spacing.

The first refinement of style is to separate it from the structure using some of the code reuse and modularization mechanisms of XSL – the named attribute set and the included file. The stylesheet in its final form, statement.xsl, uses the XSL include mechanism to bring in style attribute set definitions from the file, statement.style.xsl (see Listings 3 and 4; the complete code listings for this article are available at www.sys-con.com/xml/sourcec.cfm). Naming and separating style attributes in this way has a number of benefits. First, it helps make more evident the structure of the result document from a reading of the stylesheet. The structure is clearer because it isn't interlaced with lots of properties to affect the style.

Second, separation of the style ele-

AUTHOR BIO

Douglas Lovell is a software engineer for the IBM T. J. Watson Research Center. He has written many tools for working with XML and XSLT and a partial implementation of XSL-FO. Doug authored the XSL Formatting Objects Developer's Handbook *published by Sams in 2002.*

ments allows a separation of concerns. A graphic designer or layout artist can work on the style attributes to affect a pleasing appearance without having to meddle with the more algorithmic, programming-oriented structural transform component of the stylesheet. Finally, using named attribute sets ensures consistency whenever the stylesheet calls upon them from more than one place.

The primary structural refinements to the initial stylesheet work to wrestle the transaction and summary data into tabular form. Writing XSL templates to produce tables requires attention to a number of details. First there is the structure of the table – the division of data into cells within rows and columns. Then there are table headers and footers; column widths; table, cell, row, and column borders; and alignments; all of which must be just right before the stylesheet yields a presentable result.

There are a number of tricks that will help you get a grip on the table output of your stylesheet. The first is to set aside the computer keyboard and make a sketch. The sketch does not have to look exactly like the result table, but it should clearly show the structure. Mark off columns and label them with their headers. Write the names of the elements that will appear in each column. This sketch will help you keep the overall structure as a reference while you navigate the details of writing the table layout. The sketch shown in Figure 2 is an example.

With the sketch as a guide you can quickly fill in the basic structure of the result table. The coding for the transactions table, for example, consists of a template to match the transaction-list element paired with a template that matches each transaction element. The template that matches the transaction list outputs the table itself. The template that matches the transactions outputs one table row per transaction. Listing 2 shows the way the code appears in its initial, rough form.

This gets you started quickly, and very nicely, with the structure of the table. Now you can add the table header and footer in the base template, the one that generates the table. After that you can add column definitions, then tweak the cell alignments, borders, and so forth to get an acceptable looking report. Listing 3 gives the fully developed stylesheet and Figure 3 shows the result.

You may gain a lot of insight about how to make things work with XSL by studying the stylesheet available for download from the ***XML-J*** Web site. The transaction summary portion of the stylesheet demonstrates a few tricks in particular. First, the table layout attribute set omits table footers at line breaks. This prevents the summary total from appearing at the bottom of the first page, before the report has shown all of the transac-

STC Cust Rollover IRA Peter P Principal 23 Polity Place Perkins Pennsylvania 21203-9345 Marvin M. Moody Mandalay Mines Ltd 12 Market Mile Mint Minnessota 83128-7423 mmoody@mmines.com

Portfolio Holdings
Money Market Funds 703.62 Martin Vales Mutual Trading Corporation Federal Government Obligations, Class C 16155.449 8.61 139098.41 Reinvest Cash 6.07 8439 Fairview Government Trust Federal Government Obligations, Class C 2443.838 6.97 17033.55 Cash Cash 4.61 785 Fairview Floating Rate Corporate Obligations, Class A 2784.408 8.11 22581.54 Reinvest Cash 5.21 1176
Contribution and Distribution Summary
4905.00 13065 545.00 1965.00

Figure 1 • Statement

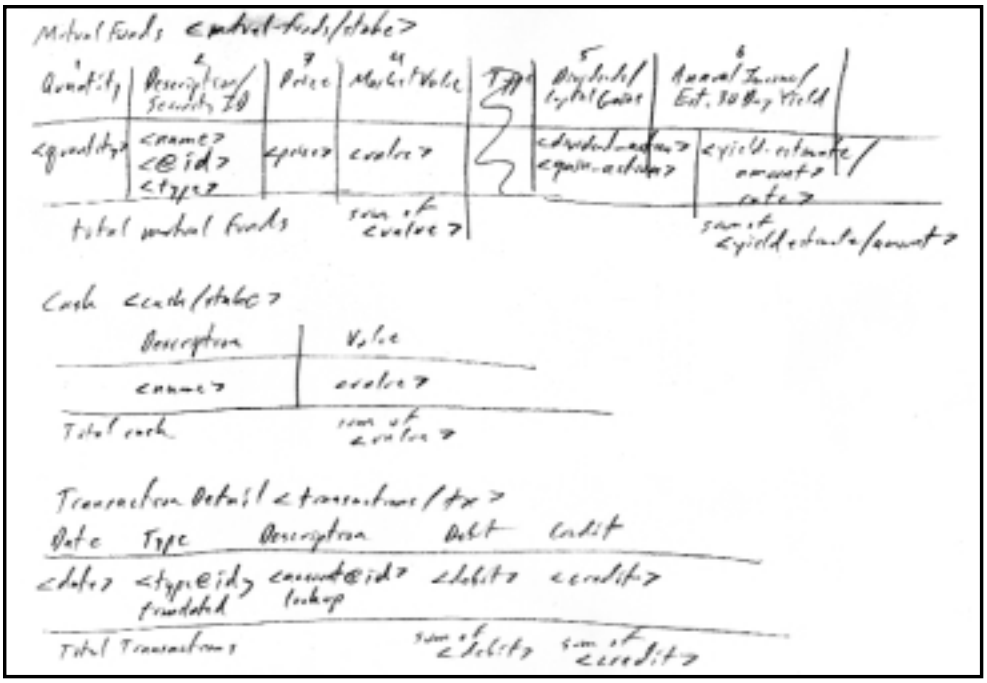


Figure 2 • Elements sketch

Mandalay Mines LTD
Your account: 4329-7723

Peter P Principal
23 Polity Place
Perkins, Pennsylvania
21203-9345

Portfolio Holdings

Cash and Equivalents

Description	Current Value
Money Market Funds	703.62
Total Cash and Equivalents	**703.62**

Mutual Funds

Quantity	Description/ Security ID	Price	Market Value	Dividends/ Capital Gains	Annual Income/ Est. 30 day yield
16,155.44900	Martin Vales Mutual Trading Corporation MVGOX	8.61	139,098.41	Cash Reinvest	8,439.00 6.07%
2,443.83800	Federal Government Obligations, Class C Fairview Government Trust FXUOX	6.97	17,033.55	Cash Cash	785.00 4.61%
2,784.40800	Federal Government Obligations, Class C Fairview Floating Rate FXFTX	8.11	22,581.54	Cash Reinvest	1,176.00 5.21%
Total Mutual Funds	Corporate Obligations, Class A		**178,713.50**		**10,400.00**

Transaction Detail

Date	Type	Description	Debit	Credit
12/02/2002	Cash Dividend	Fairview Floating Rate Corporate Obligations, Class A		95.11
12/03/2002	Purchase	Money Market Funds	95.11	
12/04/2002	Cash Dividend	Fairview Government Trust Federal Government Obligations, Class C		65.49
12/05/2002	Purchase	Money Market Funds	65.49	
12/17/2002	Withholding		545.00	
12/17/2002	Withdrawal		4905.00	
12/17/2002	Cash Dividend	Money Market Funds		0.95
12/17/2002	Sale	Money Market Funds		5449.29
12/18/2002	Purchase	Money Market Funds	0.24	

Your account manager: Marvin M. Moody, Mandalay Mines Ltd, 12 Market Mile, Mint, Minnessota 83128-7423 email: mmoody@mmines.com

Figure 3 • The result

tions. First, the stylesheet uses the sum() function. Cells that contain dollar amounts call the format-dollar-amount template to get separators at multiples of one thousand and to always print trailing zeros to show hundredths of a dollar. The format-dollar-amount template uses the format-number() function of XSL. The quantity column of the mutual funds table uses the format-number() function as well.

One difficult problem with the transactions table is to get a continuation indicator into the table header when the table prints across a page break. The use of the FO marker and retrieve-marker elements should solve this problem. The FO implementation used to produce the PDF may not support the marker element. The idea is to retrieve the marker in the header. When the header outputs for the first time the marker is empty. Table rows define the marker with the continuation text. When the header prints on subsequent pages it should contain the redefined continuation marker.

The last trick in the transactions table is the method used to retrieve the text for the description column. The template first defines a variable for that cell, named acctID, that contains the value of the ID attribute on the account. The expression, <xsl:value-of select="//stake[@id=$acctID]/name"/> retrieves the name of the account from the stake element earlier in the report. It would also be reasonable to read this information from a separate document using the document() function. The separate document would contain name, type, and other information about various investment offerings indexed by their identifiers.

Conclusion

XSL provides a potent vehicle for presenting statements and reports online and on demand. It enables separation of the report query logic from the presentation logic. It provides a unified mechanism for presenting the statements online and in print form. The online presentation provided by XSL-FO surpasses the quality enabled by HTML and equals that afforded by printed copies delivered in the mail, which should help customers feel more comfortable about forgoing mailed statements for those available online.

Resources

The PDF outputs shown in this article were produced with an evaluation copy of the Antenna House XSL Formatter version 2.3 available from www.antennahouse.com. Information about the IBM InfoPrint XML Extender for z/OS is available from www.printers.ibm.com/R5PSC.NSF/Web/xmlextenderhome. The W3C XSL Web site, www.w3.org/Style/XSL, contains links to many other implementations. The site also contains links to tutorial materials and to FO interest mailing lists. Books and articles by this author, Dave Pawson, G. Ken Holman, Michael Kay, W. Eliot Kimber, and others are all helpful as well.

DCL@US.IBM.COM

LISTING 1 • statement1.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/
  Transform"
 xmlns:fo="http://www.w3.org/1999/XSL/
  Format">

<xsl:template match="/">
  <fo:root>
    <fo:layout-master-set>
      <fo:simple-page-master
       master-name="page"
       page-width="11in"
       page-height="8.5in"
       margin-top="0.675in"
       margin-bottom="0.675in"
       margin-left="0.375in"
       margin-right="0.375in">
        <fo:region-before
         region-name="header"
         extent="1.5in"/>
        <fo:region-after
         region-name="footer"
         extent="0.675in"/>
        <fo:region-body
         margin-top="1.5in"
         region-name="body"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence
     master-reference="page">
      <fo:static-content
       flow-name="header">
        <xsl:apply-templates
select="statement/account"/>
      </fo:static-content>
      <fo:flow
       flow-name="body">
        <xsl:apply-templates/>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="account">
  <fo:block>
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="statement">
  <xsl:apply-templates select=
    "holdings"/>
  <xsl:apply-templates select=
    "transactions"/>
  <xsl:apply-templates select=
   "edge-summary"/>
</xsl:template>

<xsl:template match="holdings">
  <fo:block>
    Portfolio Holdings
  </fo:block>
  <fo:block>
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="transactions">
  <fo:block>
    Transaction Detail
  </fo:block>
  <fo:block>
     <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="edge-summary">
  <fo:block>
    Contribution and Distribution
     Summary
  </fo:block>
  <fo:block>
     <xsl:apply-templates/>
  </fo:block>
</xsl:template>

</xsl:stylesheet>
```

LISTING 2 • Initial transactions table templates

```
<xsl:template match="transaction-
list">
  <fo:block xsl:use-attribute-
sets="title-block">
    Transaction Detail
  </fo:block>

    <fo:table>
      <fo:table-body>
      <xsl:apply-templates/>
      </fo:table-body>
    </fo:table>
</xsl:template>

<xsl:template match="transaction">
  <fo:table-row>
    <fo:table-cell>
      <fo:block>
        <xsl:value-of select="date"/>
      </fo:block>
    </fo:table-cell>
    …
  </fo:table-row>
</xsl:template>

    <fo:table-cell
     xsl:use-attribute-sets=
      "numeric-cell">
      <fo:block>
       <xsl:call-template name=
        "format-dollar-amount">
         <xsl:with-param
            name="amount">
           <xsl:value-of
             select="ytd"/>
         </xsl:with-param>
       </xsl:call-template>
      </fo:block>
    </fo:table-cell>
</xsl:template>

<xsl:template match="period">
  <fo:block>
    <xsl:apply-templates
select="start"/>
    <xsl:text> to </xsl:text>
  </fo:block>
  <fo:block>
    <xsl:apply-templates
select="end"/>
  </fo:block>
</xsl:template>

</xsl:stylesheet>
```

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

International Web Services Conference & Expo

Web Services Edge WEST 2003

web services conference EDGE &expo

SEPT. 30 - OCT. 2, 2003

Santa Clara, CA

EXTENDING THE ENTERPRISE WITH WEB SERVICES THROUGH JAVA, .NET, WEBSPHERE, MAC OS X AND XML TECHNOLOGIES

XML

WebSphere

Microsoft .net

Mac OS X

Featured technologies and topics will include:

- Focus on XML
- Focus on .NET
- Focus on Java
- Focus on WebSphere
- Focus on Mac OS X

LINUX EDGE conference&expo

web services EDGE conference&expo

BOSTON

February 24-27, 2004

For more information visit
www.sys-con.com
or call
201 802-3069

Over 100 participating companies will display and demonstrate over 300 developer products and solutions.

Over 2,000 Systems Integrators, System Architects, Developers, and Project Managers will attend the conference expo.

Over 60 of the latest sessions on training, certifications, seminars, case studies, and panel discussions will deliver REAL World benefits, the industry pulse and proven strategies.

Contact information: U.S. Events: 201 802-3069 or e-mail grisha@sys-con.com

WebServices JOURNAL | JAVA DEVELOPER'S JOURNAL | WebSphere DEVELOPER'S JOURNAL | XML JOURNAL | .NET DEVELOPER'S JOURNAL | SAMS

WebLogic DEVELOPER'S JOURNAL | wireless BUSINESS & TECHNOLOGY | LINUX BUSINESS & TECHNOLOGY | CF Advisor | ColdFusion Developer's Journal | PowerBuilder DEVELOPER'S JOURNAL

WEB SERVICES EDGE WEST 2003
CALL FOR PAPERS NOW OPEN
Submit your papers online at:
www.sys-con.com/webservices2003west

PRODUCED BY SYS-CON EVENTS

Database-Driven Charting Using XSLT and SVG

Controlling finer details of data-driven Scalable Vector Graphics

WRITTEN BY
AVINASH MOHARIL & RAJESH ZADE

Over the last few years, there have been significant developments in the XML world. Many enterprises have adopted XML for various purposes such as data transfer, reporting, client-tier presentation, server-to-server communication, and so on.

The need for representing data in XML form is greater than ever. Almost all database vendors have ways of extracting data in XML form. XML technology has enabled us to drive enterprise reports based on legacy databases. A previous article, "Using FOP for Industrial Needs" (***XML-J***, Vol.3, issue 10), discussed how to generate PDF reports based on XML data. Many times, reports are not complete without good graphical presentation. This article discusses how you can generate Scalable Vector Graphics (SVG) based on XML data. You can combine the FOP and SVG processes to generate complete reports.

Scalable Vector Graphics is a markup language for describing two-dimensional graphics in XML. It is a language for presenting rich graphical content, and it allows you to create graphics from XML data using XSLT. Most of the modern devices are raster-oriented, so it comes down to where graphics are rasterized – at the client or server level. SVG is rasterized at the client level, giving more flexibility for presenting graphics. SVG gives the user full control over the rasterization process. SVG documents define graphics as vector graphics rather than a bitmap so you can scale and zoom in and out without losing any detail or blurring the image.

SVG uses a "painters model" of rendering, meaning that paint is applied in successive operations to the output device such that each operation paints over some area of the output device. When the area overlaps a previously painted area, the new paint partially or completely obscures the old.

SVG 1.1, SVG 1.0, and SVG Mobile Profiles are W3C Recommendations. Work continues on SVG 1.2 and future profiles for mobile and printing. The complete Scalable Vector Graphics (SVG) 1.1 specification can be found at www.w3.org/TR/SVG11. A full list of implementations is available from the W3C pages at www.w3.org/Graphics/SVG/SVG-Implementations.htm8.

Why XSLT and SVG?

With the help of XSLT, XML data can be transformed into SVG graphics. SVG drawings can be dynamic and interactive, which gives tremendous flexibility when building data-dependent graphics such as charts. The Document Object Model (DOM) for SVG, which includes the full XML DOM, allows for straightforward and efficient vector graphics animation. A set of event handlers such as OnMouseover and OnClick can be assigned to any SVG graphical object. Because of the compatibility of SVG with other technologies, features like scripting can be built on SVG elements and other XML elements from different namespaces simultaneously within the same Web page.

Simple SVG Application

To understand how graphics are represented in SVG, let's look at the following simple SVG document that draws a rectangle. To view a stand-alone SVG, you need to download the SVG plug-in from www.adobe.com/svg. Once you have the plug-in installed you can view graphics from any SVG file with a .svg extension in a browser such as IE5 (see Listing 1).

When the document shown in Listing 1 is loaded in IE, you should see a red rectangle with a black border (see Figure 1).

SVG uses the namespace http://www.w3.org/2000/svg to distinguish between elements that are part of SVG and elements that are part of other markup languages.

The parent document element of an SVG document is an <svg> element. Children of this element are the other elements that are used to construct an image. In the example we're drawing a rectangle, a basic shape provided by SVG, using the <rect> element. The six basic shapes supported by SVG are <rect>, <circle>, <ellipse>, <line>, <polyline>, and <polygon>.

Putting It to Real Use

Before we jump into building a demo application, look at Table 1 to see the application areas in which we can use this technology.

Case Study

Let's look at a Sales Report for National Car Dealer, Inc., for quarterly sales of various vehicles sold in all four quarters of year 2002. Following are the sales figures for all the quarters:

- ***Quarter 1 (Q1):*** $400,000
- ***Quarter 2 (Q2):*** $600,000
- ***Quarter 3 (Q3):*** $575,000
- ***Quarter 4 (Q4):*** $800,000

In this case study we'll generate a dynamic bar chart based on the above facts. Let's assume that the data is stored in the database. This article does not delve into XML data file generation, but you can easily write a DataProcessingBean and have a simple getDataAsXML() method to achieve this. Alternatively, you can simply write a SQL query to return database data as XML.

There are three components needed to generate a dynamic SVG:

- XML data file
- XSLT stylesheet
- SVG file

In addition, you'll have to download the Adobe SVG Viewer 3.0 and install it as a plug-in component for IE. XSLT is used to generate appropriate SVG elements to produce the bar chart graphic based on the dynamic data residing in an XML file. This would be the standard process for any enterprise graphic generation where enterprise data needs to be embedded in the SVG. You can simply double-click the SalesChart.svg file and IE will process it and display the graphic once you've installed the SVG viewer.

Figure 2 shows how our data processing looks.

XML data file

XML data can be generated using SQL commands (MS SQL and many other databases allow you to select rows as XML), or you can write simple Java components to generate XML files. Often, the XML data file will not represent a one-to-one relationship to the database table rows; rather, it will represent the processed data in an application tier (memory resident) based on various sources. Therefore, it is advisable to build a simple XML-generation Java object layer inside an application tier (see Listing 2).

The data shown in Listing 2 is in XML form and describes the Quarterly sales element that has elements for each quarter and the sales figures (in thousands) for each quarter. We'll use this data to draw a bar chart. The maximum sales amount in any quarter, as mentioned in our XML data file, is 900K. We can calculate this amount from data, but for our case study purpose we'll assume we have it in an XML data file.

XSLT stylesheet

XSLT is used to describe the structural transformation of XML. Generally, it is used in conjunction with XPath. XPath is used to address XML to select nodes for processing to define conditions for the way nodes will be processed and to generate text to be inserted in the resultant XML tree. In this case study the XML file is quite small and we select all of the nodes, but in a typical application you'd have a much more complex XML document to be processed using XSLT. As mentioned earlier, XPath can generate text to be inserted in the resultant XML tree. In this case the text to be inserted using XSLT is the SVG-specific text that would wrap the dynamic XML data and generate a well-formed resultant XML document that conforms to SVG specifications. As long as the resultant XML file is a valid SVG document, you can process it using an SVG viewer, such as one from Adobe, and generate the final graphic.

The main parts of our SalesChart XSLT document are:

- Define SVG window properties
- Define static graphic properties
- Define dynamic graphic properties

Below are the parts of the SalesChart XSLT document that relate to the above points.

```
<!-- Size of the SVG window -->
  <svg width="600" height="500">
 . . .
```

Industry	Use
Banking, Investment, and Finance	Charts are widely used in the banking and investment sectors to effectively communicate important information. SVG can be used to represent that information on the fly.
General Navigation and Topography	Complex and dynamic navigational maps can be represented as SVG. Topographical data can be represented as SVG.
Custom Products	Most highly configurable products need to be assembled on the fly using various components. Components of the products can be represented as individual SVG graphics and can be scaled up or down depending on the need.
Retail	ERP systems can effectively use SVG to represent data graphically.
Government	Various reports can be generated using XSLT-SVG techniques to enhance report data.
Utilities	SVG can be effectively used to represent various usage trends.

Table 1 • Application areas

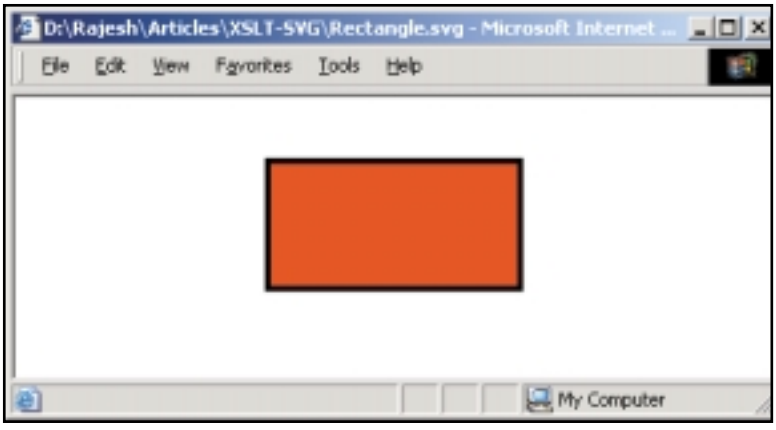


Figure 1 • Graphic from SVG file

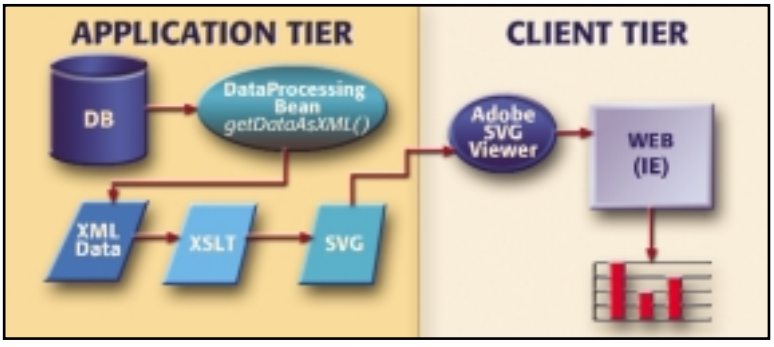


Figure 2 • Generating charts using XSLT and SVG

Figure 3 • SVG rendered in IE

SVG element attributes define graphic window width and height. We calculate the graphic properties using various elements such as <text>, <rect>, <line>, and <g>. Here we add header information, generate gray areas, and draw vertical and horizontal lines (see Listing 3).

The next step is building a dynamic graphic. We're trying to build a bar chart that reflects the quarterly sales figures. To draw the chart with relevant dimensions we need to consider the height of the tallest bar, the number of bars, the calibration of the *Y*-axis (bar height), and the spreading of bars evenly over the *X*-axis. These properties are very graphic-specific and will vary based on what you're trying to build. Nonetheless, this gives you an idea of how to build a dynamic graphic using SVG elements (see Listing 4).

Building this particular graphic is an iterative process. We start by matching the template on the root element (please see the complete code listing, available at www.sys-con.com/xml/sourcec.cfm). Then we iterate for each quarter by matching a template on the "Quarter" element. In the listing we're constructing a bar using a rectangle based on dimensions derived from quarterly figures.

The graphic is placed on the canvas using a coordinate system. The top left corner of the canvas is (0,0) point and the value increases as you move to the right or to the bottom. The bar height is calculated using the sales of each quarter. The maximum bar height for our graphics is 300 units. The maximum sales revenue for any quarter is $900K. We calculate the height of the bar for each quarter using the formula barHeight= ($sales/900) * 300 units. Based on the number of bars and the bar height we determine the *X* and *Y* positions for the vertical bars. The width of the bar is fixed at 40. The distance between the bars is fixed at 40. For example, we spare 100 units for the left margin and 40 units for space between the vertical axis and our first vertical bar. So the *X* coordinate on our coordinate system should be 140 – the equivalent of 140 units from the left border on canvas. We know that the *Y* coordinate for our *X*-axis is 400 as we have drawn a horizontal line from point (100,400) to point (500,400). In order to determine the *Y* coordinate of the left top corner of our bar rectangle, we need to subtract the height of the bar from 400. For our first bar using the above formula, the barHeight is 150, so we need to draw a rectangle with (*X*,*Y*) as (140,250), width 40 and height 150. Effectively, we are drawing a rectangle with points (140,250), (180,250), (140,400), (180,400). The bottom line of our bar rectangle merges on the *X*-axis. In order to write a label for our bar, we write "Q1" text just below our *X*-axis at (150,420). For drawing subsequent bars for other quarters we calculate the (*X*, *Y*) positions in a similar fashion.

SVG file

Once you process the above XSLT file on the input XML document, it produces an XML document that is also a well-formed SVG document (see Listing 5).

Figure 3 shows how the final graphic is displayed in IE when SalesChart.svg file is loaded in IE.

Advanced SVG Techniques

SVG supports the ability to change vector graphics over time. Animation has been very popular in electronic media. SVG's animation elements were developed in collaboration with the W3C Synchronized Multimedia (SYMM) Working Group, developers of the Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. For detailed information about SVG animation and related elements, see the specification at www.w3.org/TR/SVG.

SVG can also be used in XSL-FO (Formatting Objects) to use graphics effectively in print media. Listing 6 demonstrates how you can include the above chart graphics in a PDF document.

Conclusion

The effective use of XSLT and SVG can help build dynamic and scalable graphics for data presentation. These technologies can be used to present raw XML data in more visual forms like charts and graphs for Web and print media. Since XML documents can be easily generated based on enterprise data, SVG becomes a truly dynamic graphic based on enterprise data.

Resources

- *Code for this article:* www.sys-con.com/xml/sourcec.cfm
- *XSLT specifications:* www.w3.org/TR/xslt
- *SVG specifications:* www.w3.org/TR/SVG
- *XMLSPY:* www.xmlspy.com
- *Adobe SVG viewer plug-in:* www.adobe.com/svg

Author Bios

Avinash Moharil has more than 14 years of experience in the computing field and has been working as a senior technology consultant and project manager in Silicon Valley.

Rajesh Zade has over 14 years of experience in the computing field and has been working in various Java and e-commerce technologies since 1996. Currently, he works as a chief technical architect for NetCliff, Inc., based in Santa Clara, CA.

RAJESH.ZADE@NETCLIFF.COM

AVINASH.MOHARIL@NETCLIFF.COM

LISTING 1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/REC-SVG-
  20010904/DTD/svg10.dtd">
<svg width="12cm" height="5cm" viewBox="0 0 1200 500"
  xmlns="http://www.w3.org/2000/svg">
 <rect x="400" y="100" width="400" height="200" fill="red"
  stroke="black" stroke-width="10"/>
</svg>
```

LISTING 2

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="D:\Rajesh\
  Articles\XSLT-SVG\SalesChart.xsl"?>
<QuarterlySales>
 <Year>2002</Year>
 <MaxValue>900</MaxValue>
 <Quarter name="Q1">450</Quarter>
 <Quarter name="Q2">600</Quarter>
 <Quarter name="Q3">575</Quarter>
 <Quarter name="Q4">800</Quarter>
</QuarterlySales>
```

LISTING 3

```
<!-- Heading for our chart -->
   <text x="100" y="50" style="font-size:18;
      font-weight:normal; stroke:blue; fill:blue;">
    National Car Dealer - Quarterly Sales <xsl:value-
      of select="/QuarterlySales/Year"/>
   </text>
   <text x="220" y="70" style="font-size:14;
      font-weight:normal; stroke:red; fill:red;">
    (Values in Thousands) <xsl:value-of select="/
       QuarterlySales/Year"/>
   </text>
   <!-- Draw our gray area -->
   <rect x="100" y="90" width="400" height="310"
      style="fill:#CCCCCC;"/>
   <!-- Draw the vertical line up the left of the gray
          area y-axis -->
   <line x1="100" y1="90" x2="{100}" y2="{400}"
    style="stroke:#000099; stroke-width:0.1"/>
   <!-- Draw the horizontal lines -->
   <g style="fill:#000000; font-size:12;
    font-family:Arial">
    <text x="85" y="{400}">
     <xsl:value-of select="0"/>
    </text>
    <line x1="100" y1="400" x2="{100 + 400}" y2="
     {400}" style="stroke:#000000; stroke-width:0.2"/>
   </g>
```

LISTING 4

```
<xsl:template match="/">
  <svg width="600" height="500">
   . . .
   <xsl:apply-templates select="//Quarter"/>
  </svg>
 </xsl:template>
 <!-- Create a vertical bar and horizontal labels for
  each <Quarter> element in the XML source document -->
 <xsl:template match="/QuarterlySales/Quarter">
  <xsl:variable name="NumofBar">
   <xsl:number count="*"/>
  </xsl:variable>
  <xsl:variable name="Sales">
   <xsl:value-of select="."/>
  </xsl:variable>
  <xsl:variable name="BarHeight">
   <xsl:value-of select="$Sales * 300 div 900"/>
  </xsl:variable>
  <!-- Vertical Bar -->
  <rect x="{140 +(40 * 2 *($NumofBar - 3))}" y="{400 -
  $BarHeight}" style="fill:red; stroke:black;
   stroke-width:0.1" width="{40}" height="{$BarHeight}"/>
  <!--Horizontal labels -->
  <text x="{150 +(40 * 2 *($NumofBar - 3))}" y="420">
   <xsl:value-of select="@name"/>
  </text>
 </xsl:template>
```

LISTING 5

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="600" height="500">
 <text x="100" y="50" style="font-size:18;
  font-weight:normal; stroke:blue; fill:blue;">
    National Car Dealer - Quarterly Sales 2002</text>
 <text x="220" y="70" style="font-size:14;
  font-weight:normal; stroke:red; fill:red;">
    (Values in Thousands) 2002</text>
 <rect x="100" y="90" width="400" height="310"
   style="fill:#CCCCCC;"/>
 <line x1="100" y1="90" x2="100" y2="400"
   style="stroke:#000099; stroke-width:0.1"/>
 <g style="fill:#000000; font-size:12; font-family:Arial">
  <text x="85" y="400">0</text>
  <line x1="100" y1="400" x2="500" y2="400"
    style="stroke:#000000; stroke-width:0.2"/>
 </g>

 . . .
 <rect x="140" y="250" style="fill:red; stroke:black;
   stroke-width:0.1" width="40" height="150"/>
 <text x="150" y="420">Q1</text>
 <rect x="220" y="200" style="fill:red; stroke:black;
   stroke-width:0.1" width="40" height="200"/>
 <text x="230" y="420">Q2</text>
 <rect x="300" y="208.33333333333334" style="fill:red;
   stroke:black; stroke-width:0.1" width="40"
  height="191.66666666666666"/>
 <text x="310" y="420">Q3</text>
 <rect x="380" y="133.33333333333331" style="fill:red;
   stroke:black; stroke-width:0.1" width="40"
  height="266.66666666666669"/>
 <text x="390" y="420">Q4</text>
</svg>
```

LISTING 6

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="my-page">
      <fo:region-body margin="1in" />
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-reference="my-page">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>
            <fo:external-graphic src="SalesChart.svg" />
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

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

JavaOne

java.sun.com/javaone/sf

JavaOne

java.sun.com/javaone/sf

WRITTEN BY **BING LI AND WEI-TEK TSAI**

Introduction to LBXML Operator

A Java API-based tool for XML insertion, modification, searching, and removal

LBXML Operator is a Java API-based XML tool that supports insertion, modification, searching, and removal on XML files. Of the many available XML parsers and XML tools, none provide APIs to complete the above functionalities when manipulation conditions become complex – they support only simple operations on XML.

With LBXML Operator, after specifying conditions using Java-rich data structures, any particular value between tags in an XML file can be touched and returned as Java data structures.

Application Situations

With the development of the Internet and distributed technologies, XML has become more and more important in data representation, automatic data exchange over heterogeneous platforms, data modeling, search engines, and even data storage. When developers use XML it is essential that they be able to access values between tags. XML is usually used to describe complex data structures, and being able to manipulate XML makes developers happy – LBXML Operator is a good tool for this.

Representing data in XML

Before XML, developers had to represent data in plain text format. To describe data, they had to define their own format using special symbols like "|", "#", and so on. Plain text isn't standard and can be understood only by the developer who defines it. Without proper comments, even the developers themselves forget the sense of the format after some time. XML eliminates this nightmare – developers can define meaningful tags and put corresponding values between them. Using rich XML data structures, they can represent data in a human-readable format and express a variety of senses.

However, developers may need to access XML files. For example, if an XML file is a configuration file, after the program starts to run it has to read some initial data from it. At that time, LBXML Operator can be used to retrieve data from the configuration file in XML. Some information in the configuration file may need to be updated after the program runs. LBXML Operator is a tool to modify the corresponding data in the configuration file.

Data storage

Another advantage XML gives developers is the ability to store data in XML, temporarily or permanently. Frequently a lot of initial data is required for a program to start up, or a great deal of output data is generated while the program is running. A good solution is to store all the data in XML files temporarily, which decreases the number of visiting databases. Sometimes, if the system capability allows or the amount of data is not too large, it's feasible to store the data in XML files instead of databases.

In this case it's necessary to manipulate XML files just like you access databases. LBXML Operator provides Java APIs, which are similar to SQL statements, for developers to access data in those XML files.

Automatic data exchange

The most significant use of XML is in data exchange between heterogeneous platforms. Data in the exchanged XML files must be processed; if this is automatic, it is really beneficial. What's needed is a program that can access XML files flexibly and powerfully. Since LBXML Operator is a collection of Java APIs, developers can program with those APIs to design an interface capable of processing data exchange in XML.

Search engines

More and more data over the Internet is described in XML rather than HTML, since XML can express much richer information than HTML. To build a good search engine, developers need a tool that gives them a hand – LBXML Operator provides APIs that can be used to touch any values between any tags in XML files.

Data modeling

Another important use of XML is in describing data models. For example, in the area of e-commerce developers and scientists describe business process models in XML. In the area of workflow management developers describe workflows in XML. In short, data models can be represented in XML in practically all areas.

Usually, those data models drive a system to work. LBXML Operator APIs provide runtime support for developers to access data models in XML.

Existing Java-Based XML Tools

The major difference between LBXML Operator and existing Java-based XML tools is the approach to specifying manipulation conditions. Most existing tools define such conditions using XQuery, XPath, or their own script languages. However, LBXML Operator depicts conditions through APIs, i.e., a particular API handles a specific case to manipulate XML files.

One major advantage to specifying manipulation conditions using XQuery, XPath, or script languages is that those descriptions have standard specifications and cover all situations of XML searching

AUTHOR BIO

Bing Li is a PhD student at Arizona State University, focusing on the research of e-commerce/e-business. In the summer of 2002, Bing Li worked with research staffs at the IBM T.J. Watson Research Center. In 1998 and 1999, he worked for Bell Labs and Motorola Research Center in China.

– basic query, range query, Max/Mix, and so on. LBXML Operator searching includes basic query only. Furthermore, existing tools search XML based not only on tags but also on attributes. Searching with the latest version of LBXML Operator is based on tags only.

The major disadvantage of existing tools is that each covers only one type of manipulation. Some tools are used for searching and some are used for modification. LBXML Operator supports all the fundamental operations – insertion, modification, searching, and removal.

Another disadvantage of existing tools is that they regard XML files as tagged plain text files; therefore, they don't utilize an important feature of XML, hierarchy. LBXML Operator views XML files as structured plain text files. An XML file is organized by similar structured plain text, much like a record in a relational database. A unique key or multiple keys exist in each XML file; thus, manipulating XML files with LBXML Operator is like manipulating records in a table of databases.

With existing tools, developers have to write script languages that are separated from code, which requires extra effort. However, LBXML Operator utilizes rich Java data structures to specify operation conditions. Developers don't have to spend extra time on script languages except Java code. In addition, existing tools don't return results of manipulations in Java-rich data structures. Most result in an XML document, which is not convenient for developers. LBXML Operator results are put into Java-rich data structures, such as Hashtable, HashSet, and String, which are easy for developers to program.

Overall Approach

LBXML Operator is implemented for Java programmers. Since XML is applied widely in the current Internet world, and Java is the first choice for Internet programmers, LBXML Operator is a good candidate for those Web developers who program with Java and XML.

Features

LBXML Operator has two important features. First, LBXML Operator regards XML files as hierarchical structure–based plain text files. Each XML file consists of a series of similarly structured sub-XML files and has its own keys. This architecture looks like that of a relational database. Based on this understanding, LBXML Operator provides a collection of APIs that manipulate XML just like SQL statements for tables in databases.

The second important feature is the goal of LBXML Operator – to provide convenient XML tools for programmers that are compatible with a particular programming language, i.e., Java. Programmers can specify operation conditions through Java data structures, manipulate XML with Java APIs, and get results in Java data structures.

Concepts Used

Tag and value, updating tag, parent tag, key tag, and sibling tag – all these are defined when processing XML files using LBXML Operator. They describe conditions that determine which tag's value should be accessed in an XML file, working like navigators to help LBXML Operator find the appropriate tag and then operate on it.

Tag and its value

When performing an operation on an XML file, tag and its value are two issues that are always taken into account. In an XML file, some tags have attached values and some tags don't have any values. By default, all the operations provided by LBXML Operator are concentrated only on tags that have attached values. LBXML Operator regards the combination of a tag and its value as a basic node or unit of XML, which is different from the view of XML parsers. From the view of XML parsers, a tag and its value are considered two separate nodes.

Updating tag

Updating tag specifies the tag whose value is accessed by LBXML Operator, and consists of operations such as insertion, modification, searching, and removal.

Key tag

This concept is borrowed from the domain of relational databases. In each table of a relational database, there is at least one field that behaves as a key. SQL statements identify each row in a table based on the key field, which is the field through which a row in the table is different from others. Either a unique field or multiple fields can form a key for a table.

Similarly, a particular tag can be defined as a key tag for a specific structure in an XML file. In general, an XML file represents information in hierarchical structures and each XML file is organized by similar multiple structures. Each structure has one or more tags whose values differentiate the structure from others. Such tags are called key tags for the particular structure.

For example, in Listing 1 <Version> is the key tag for the entire XML file. <Organization> is the key tag for the structure between <SAT> and </SAT>. For the structure between <Form> and </Form>, the key tag is <Request>. <Organization>, <User>, and <Workflow> are also key tags for the structure between <SAT> and </SAT> since their values differentiate the structure between <SAT> and </SAT> from other structures between <SAT> and </SAT>.

According to the definition of key tag, each key tag has value. A tag without a value is not a key tag in LBXML Operator.

Parent tag

Since an XML file can be interpreted as a tree (DOM) structure, each tag resides in a node of a tree. Thus, a tag and its value must have different levels. If tag A is one level higher than tag B, tag A is the parent of tag B. For example, in Listing 1 <NewSAT> is the parent tag of <SAT>. <Form> is the parent tag of <Request>.

Sibling tag

Similar to the concept of the parent tag, if tag A resides in the same level as tag B, tag A is the sibling tag of tag B. For example, in Listing 1 <Request> is the sibling tag of <Response>. Two sibling tags can be the same – in Listing 1, between <Vocabulary> and </Vocabulary> are two <VocabularyName> tags. Since those two reside in the same level of the DOM tree, they are sibling tags to each other.

ParentSibling tag

If tag A is the parent tag of tag B and tag C is the sibling tag of tag A, tag C is the ParentSibling tag of tag B. In Listing 1, <Vocabulary> is the parent of <VocabularyName> and <Organization> is the sibling of <Vocabulary>, so <Organization> is the ParentSibling tag of <VocabularyName>.

Limitations of LBXML Operator

LBXML Operator does not support XML manipulations based on attributes. This feature will be added in a future version, since attribute values are also important in representing data. In our experience all the attributes of XML tags can be converted to tags, and XML without attributes looks interesting, so we suggest XML developers use tags instead of attributes.

Another issue is resource occupation. Since an entire XML file is loaded in memory during manipulation of XML files, it uses a great deal of memory if the XML file is

large. It's a good idea to load a large XML file into memory part by part and combine results after the entire XML file is processed.

LBXML Operator provides powerful searching and modification operations, while insertion and removal operations are limited. More APIs for insertion and removal will be added in a future release.

Case Studies

This section shows two examples of manipulating XML with LBXML Operator. The first involves modifying the value between a pair of tags and the second involves searching a value between a pair of tags. Both require specification of operation conditions in advance.

changeByKeyTagKeyValueSiblingTagUpdateTagNewValue()

For the XML file shown in Listing 1, sometimes it is necessary to modify a tag's value based on both key tags and sibling tags. For example, users would like to modify the value, *CreditRequestNo*, of <VocabularyName>.

The method *changeByKeyTagKeyValueSiblingTagUpdateTagNewValue()* is used to handle this problem. The format of the method is shown below:

```
void changeByKeyTagKeyValueSibling-
TagUpdateTagNewValue(String xmlFile,
String keyTag, String keyValue, String
siblingTag, String siblingValue, String
updateTag, String newValue)
```

For example, to change the value of <VocabularyName> from *CreditRequestNo* to *CreditNo*, use the following code:

```
LBXMLOperator lbxmlOperator = new
LBXMLOperator();
lbxmlOperator.changeByKeyTagKeyValueSi-
blingTagUpdateTagNewValue("./xmlfile.xm
l", "Organization", "RequiredCred-
itChecking.com", "VocabularyName",
"OrderedNumber", "VocabularyName",
"CreditNo");
```

After the operation, the changed XML file appears as shown in Listing 2 (Listings 2-4 are available at www.sys-con.com/xml/sourcec.cfm).

selectByMultipleTagsAndWhere()

selectByMultipleTagsAndWhere() is a powerful approach to searching. Using this method, users can specify complex conditions to retrieve the value of a tag. The format of the method is shown below:

```
String selectByMultipleTagsAndWhere
(String xmlFile, Hashtable keyTagHash,
Hashtable keyValueHash)
```

There are two Hashtables in the parameters of the method, which are used to store complex conditions to retrieve a value of a tag. The first one, *keyTagHash*, is used to store key tags and the second, *keyValueHash*, is used to store corresponding key values. With those constraints, the method is able to retrieve the value of a particular tag exactly.

The XML file shown in Listing 3 demonstrates the utilization of the method.

For example, to search the value of <ContractName>, which is underlined, *selectByMultipleTagsAndWhere()* is used. The corresponding code is shown in Listing 4. The result of Listing 4 is shown below:

```
contractName = CreditCheckingContract
```

Conclusion

I've been using LBXML Operator for more than a year, and it has worked well. One big advantage is that the tool is based totally on APIs, so developers don't have to study particular XML query languages such as XQuery and other query scripts. Through these APIs, developers specify operation conditions according to XML structures. Furthermore, XML operations are regarded much like SQL statements, which makes the tool more acceptable since SQL statements have been widely used for some time.

Some APIs are simple – they don't provide parameters to figure out tags. Some methods are complex since a particular tag is accessed through rich parameters. In general, simple methods are used to operate many tags' values at once. Complex methods are fit for accessing the value of a particular tag. Although it's possible to design more complicated APIs, these methods can deal with most cases in XML operations.

Resources

- *GMD-IPSI XQL:* http://xml.darmstadt.gmd.de/xql
- *XSet:* www.cs.berkeley.edu/~ravenben/xset
- *Fxgrep:* www.informatik.uni-trier.de/~aberlea/Fxgrep
- *Quip:* http://developer.softwareag.com/tamino/quip
- *XML:QL:* http://theoryx5.uwinnipeg.ca/mod_perl/cpan-search?dist=XML-QL

BING.LI@ASU.EDU

LISTING 1

```
<?xml version="1.0"?>
<!DOCTYPE RequirementInSAT SYSTEM "requirement_in_sat.dtd"
>
<RequirementInSAT>
<Version>2.10</Version>
<NewSAT>
<SAT>
<Organization>BigBug.com</Organization>
<User>customer</User>
<Workflow>Customer-Retailer</Workflow>
<Form>
<Request>OrderForm</Request>
<Response>ReceiptForm</Response>
</Form>
<Vocabulary>
<VocabularyName>OrderedNumber</VocabularyName>
<VocabularyName>InputCreditCardNumber</VocabularyName>
</Vocabulary>
<Contract>
<ContractName>WholesaleContract</ContractName>
<ContractName>CreditCheckingContract</ContractName>
</Contract>
</SAT>
</NewSAT>
<ExistingSAT>
<SAT>
<Organization>RequiredCreditChecking.com</Organization>
<User>guest</User>
<Workflow>Customer-Creditor</Workflow>
<Form>
<Request>CreditRequestForm</Request>
<Response>CreditResponseForm</Response>
</Form>
<Vocabulary>
<VocabularyName>OrderedNumber</VocabularyName>
<VocabularyName>CreditRequestNo</VocabularyName>
</Vocabulary>
<Contract>
<ContractName>CreditCheckingContract</ContractName>
</Contract>
</SAT>
</ExistingSAT>
</RequirementInSAT>
```

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

LBXML Operators

Tool	LBXML Operator	
Package	com.lblabs.xmltool/	
Class	LBXMLOperator	
	Constructor Summary	LBXMLOperator()
	Method Summary	
	void	insertNodeByParentTag(String xmlFile, String parentTag, String updateTag, String updateValue)
	void	insertNodeParentTagAndParentSibling(String xmlFile, String parentTag, String parentSiblingTag, String parentSiblingValue, String updateTag, String updateValue)
	void	removeNodeByTagValue(String xmlFile, String tag, String value)
	void	changeByValue(String xmlFile, String oldValue, String newValue)
	void	changeByTag(String xmlFile, String tag, String newValue)
	void	changeByTagNewValue(String xmlFile, String tag, String oldValue, String newValue)
	void	changeByNoTagNewValue(String xmlFile, String tag, int no, String newValue)
	void	changeBySiblingTagNewValue(String xmlFile, String siblingTag, String siblingValue, String newValue)
	void	changeBySiblingTagUpdateTagNewValue(String xmlFile, String siblingTag, String siblingValue, String updateTag, String newValue)
	void	changeByKeyTagKeyValueUpdateTagNewValue(String xmlFile, String keyTag, String keyValue, String updateTag, String newValue)
	void	changeByKeyTagKeyValueSiblingTagUpdateTagNewValue(String xmlFile, String keyTag, String keyValue, String siblingTag, String siblingValue, String updateTag, String newValue)
	void	changeByMultipleTagsAndWhere(String xmlFile, Hashtable keyTagHash, Hashtable keyValueHash)
Class	SelectorByKeyTag	
	Constructor Summary	SelectorByKeyTag()
	Method Summary	
	String	selectByKeyTag(String xmlFile, String keyTag)
Class	SelectorByTagAndWhere	
	Constructor Summary	SelectorByTagAndWhere()
	Method Summary	
	String	selectByTagAndWhere(String xmlFile, String keyTag, String searchTag, String keyValue)
Class	SelectorByBelowTagAndWhere	
	Constructor Summary	SelectorByBelowTagAndWhere()
	Method Summary	
	String	selectByBelowTagAndWhere(String xmlFile, String belowKeyTag, String searchTag, String belowKeyValue)
Class	SelectorForHash	
	Constructor Summary	SelectorForHash()
	Method Summary	
	Hashtable	selectHash(String xmlFile, String hashTag)
	SelectorForSet	
	Constructor Summary	SelectorForSet()
	Method Summary	
	HashSet	selectSet(String xmlFile, String setTag)
Class	SelectorByTagAndWhereForHash	
	Constructor Summary	SelectorByTagAndWhereForHash()
	Method Summary	
	Hashtable	selectByTagAndWhereForHash(String xmlFile, String keyTag, String searchTag, String keyValue)
Class	SelectorByTagAndWhereForSet	
	Constructor Summary	SelectorByTagAndWhereForSet()
	Method Summary	
	HashSet	selectByTagAndWhereForSet(String xmlFile, String keyTag, String searchTag, String keyValue)
Class	SelectorByMultipleTagsAndWhere	
	Constructor Summary	SelectorByMultipleTagsAndWhere()
	Method Summary	
	String	selectByMultipleTagsAndWhere(String xmlFile, Hashtable keyHash, Hashtable keyValueHash)
Class	SelectorByMultipleTagsAndWhereForHash	
	Constructor Summary	SelectorByMultipleTagsAndWhereForHash()
	Method Summary	
	Hashtable	selectByMultipleTagsAndWhereForHash(String xmlFile, Hashtable keyHash, Hashtable keyValueHash)
Class	SelectorByMultipleTagsAndWhereForSet	
	Constructor Summary	SelectorByMultipleTagsAndWhereForSet()
	Method Summary	
	HashSet	selectByMultipleTagsAndWhereForSet(String xmlFile, Hashtable keyHash, Hashtable keyValueHash)

WRITTEN BY **JOEL AMOUSSOU**

XML Certification Quizzer

Validating your XML skills

This column has two objectives. The first is to help you prepare for IBM Test 141 on XML and related technologies. The second is to help you learn XML by offering tips for designing and optimizing XML solutions.

The first question in your mind is probably: Why should I get XML certified? There are three simple answers to that question. First, exam preparation brings discipline and rigor to the process of learning XML and the very large collection of related specifications. Second, there is no doubt that being XML certified will make you more competitive in today's tight IT job market. For example, some government organizations make certification a prerequisite for obtaining grants and contracts for new projects. If you already have a job, certification could result in a promotion, salary increase, or better job security. Finally, the ultimate goal of learning is performance. Preparing for the exam and becoming certified will arm you with the knowledge and skills you need to architect and implement the right solution for your next XML project.

For more information on the objectives of IBM Test 141 and a sample test, visit www-1.ibm.com/certify/tests/sam141.shtml. The format of the exam is multiple choice. This column presents five questions from the five different exam objectives. It provides the correct answer and detailed explanations for each question. The IBM XML certification is not product specific. It covers W3C specifications and other vendor-neutral specifications like SAX and UDDI. The knowledge and skills acquired during exam preparation can be used with a variety of products and in different XML project scenarios.

The exam contains approximately 57 questions. The objectives are grouped into five different categories:

- ***Architecture:*** This objective represents about 19% of the exam questions. Some questions are scenario-based. Based on various business and technical considerations you will be asked to select the appropriate technologies for an XML-based solution. The choices will include XML Schema, DTDs, XSLT, CSS, DOM, SAX, Namespaces, and others. You should understand the roles of SOAP, WSDL, and UDDI in Web services architecture and how these three technologies complement each other and can be used together. It is helpful to have a basic knowledge of the structure of SOAP messages as well as the relationship between XML Schemas and WSDL. To answer XML security–related questions correctly you should know the types of data objects that can be signed or encrypted as well as the basic processing rules for signature validation and decryption, respectively.
- ***Information Modeling:*** This objective represents about 26% of the exam questions. You will be asked to translate modeling requirements into DTD and XML Schema constructs. A strong knowledge of the syntax of DTDs and XML Schemas is a requirement. This exam objective also covers the proper use of Namespaces, XLink, and XPointer.
- ***XML Processing:*** This objective represents about 33% of the exam questions. You should familiarize yourself with DOM and SAX interfaces and the methods defined by those interfaces. You should be able to write correct XPath expressions and functions to select nodes in an XML document. Other questions are related to the use of XSLT elements and functions.
- ***XML Rendering:*** This objective represents about 11% of the exam questions. It covers the use of CSS, XSLT, and XSL Formatting Objects specifications in rendering XML data. It is important to know how these technologies can be used together to render XML content and the differences in their processing models. In addition, you should have a basic understanding of the purposes of the different formatting objects.
- ***Testing and Tuning:*** This objective represents about 11% of the exam questions. You will be tested on how to optimize XML solutions using various techniques like stylesheet and schema modularization, uniqueness and referential constraints, and other constructs. The exam questions related to this objective focus on XML application testing, data query, and integration issues.

The first step in your exam preparation process is to take the IBM sample test at http://certify.torolab.ibm.com. Your test results will give you a good idea of where you should concentrate your efforts. There is a variety of resources for preparing for the exam, and you should select the one that best fits your needs and budget. Available educational resources include classroom training, e-learning (live and asynchronous), books, and online discussion groups, to name a few. Exam simulators are a very effective method for preparing for the final exam. Exam simulators include practice tests that can be taken in study mode and in exam mode. In study mode the simulator indicates the correct answer for each question and offers detailed explana-

AUTHOR BIO

Joel Amoussou is founder and chief learning architect of XMLMentor.Net, where he develops blended learning solutions for building and assessing XML skills. Joel is the author of an XML exam simulator and teaches live e-learning courses on XML certification.

tions, including tips for designing and optimizing XML solutions. In exam mode the simulator mimics the interface of a real exam. An exam simulator also gives students the ability to view questions as flashcards. It's very helpful to get some hands-on coding practice as well.

The following practice questions will give you a good idea of what to expect at the exam. The exam objective covered by each question is indicated in parentheses after the question number.

Question 1 (Architecture)

A manufacturing company is using a legacy application that imports and exports data in comma-delimited flat files. XML will be used as the data exchange format between the manufacturing company and its suppliers and customers.

Which of the following is MOST likely to be required by this architecture?

A. The element types and attribute names in the XML documents sent by suppliers and customers must match the names of data fields in the manufacturing company's legacy system.
B. Before sending documents to suppliers, the comma-delimited flat files must be transformed into XML by writing a DOM application.
C. The XML documents received from suppliers and customers must be validated and transformed into comma-delimited flat files by writing an XSLT transformation.
D. An XSLT processor will be used to transform comma-delimited flat files data into XML before sending documents to suppliers and customers.

Select two answers.

Explanation: Choice A is incorrect. The element types and attribute names used by suppliers need not be the same as the data fields in the legacy system. The data can be mapped. XML is all about interoperability between heterogeneous applications.

Choice B is correct because the DOM is a very flexible method for constructing XML document trees from scratch. The Document interface contains factory methods for creating nodes and building a DOM tree.

Choice C is correct because the result of an XSLT transformation can be XML or non-XML data.

Choice D is not correct because the input to an XSLT transformation must be a well-formed XML document. Therefore, a comma-delimited flat file cannot be processed with XSLT.

Question 2 (Testing & Tuning)

Consider the following XSLT element and function:

```
<xsl:key name="chapkey" match="
  chapter" use="@author"/>

key("chapkey", 'durand')
```

Which of the following elements will be returned by the key function?

A. <chapter chapkey= 'durand'>
B. <chapkey author= 'durand'>
C. <chapter author= 'durand'>
D. <chapter key= 'durand'>

Select one answer.

Explanation: Choice C is the correct answer. The <xsl:key> element is used to declare keys. The "name" attribute specifies the name of the key. The <xsl:key> element specifies the key of any node that matches the pattern specified in the "match" attribute. In this case, the "match" attribute is a pattern that selects <chapter> elements. Therefore, the key is applicable to <chapter> elements.

The "use" attribute is an expression specifying the value of the key. The value of the "use" attribute is an expression that returns the "author" attribute of the <chapter> element.

The key function key ("chapkey", 'durand') returns a node set containing the nodes in the same document as the context node that have a value for the key named "chapkey" equal to the string "durand". The key function will return <chapter> elements that have an "author" attribute with value "durand".

Question 3 (Information Modeling)

A DTD in a catalog.dtd file contains the following declarations:

```
<!ENTITY % discount 'IGNORE' >
<!ENTITY % regular 'INCLUDE' >
<![%discount;[
<!ELEMENT Item (Description?, Part-
Number, DiscountedPrice, Manufactur-
er+)>
]]>
<![%regular;[
<!ELEMENT Item (Description, PartNum-
ber, Price, Manufacturer?)>
]]>
```

Consider the following document type declaration in an XML document:

```
<!DOCTYPE Catalog SYSTEM
"catalog.dtd" [
<!ENTITY % regular 'IGNORE' >
<!ENTITY % discount  'INCLUDE' >
]
```

Which of the following is TRUE about the structure of an <Item> element in the XML document?

A. Inside an <Item> element, the <Manufacturer> element must occur before the <DiscountedPrice> element.
B. An <Item> element must contain at least one <Price> child element.
C. The first child element of an <Item> element must be a <Description> element.
D. An <Item> element must contain at least one <Manufacturer> child element.

Select one answer.

Explanation: Choice D is the correct answer. The declarations in a conditional section are part of the DTD only if the keyword of the conditional section is INCLUDE. The declarations in a conditional section are not logically part of the DTD if the keyword of the conditional section is IGNORE.

In this case, the keywords of the two conditional sections are parameter-entity references. The parameter-entity references must be expanded to determine whether to include or ignore the conditional sections. The parameter entities "discount" and "regular" are declared in both the internal subset and the external subset of the document type declaration. Entity declarations in the internal subset take precedence over those in the external subset.

The internal subset sets the "discount" keyword to INCLUDE. The conditional section with keyword "discount" is used to validate the XML document. The <Item> element must contain the following elements in the order specified:

1. An optional <Description> element
2. A required <PartNumber> element
3. A required <DiscountedPrice> element
4. One or more <Manufacturer> elements

Question 4 (XML Processing)

Consider the following XML document:

```
<!DOCTYPE          a [
<!ELEMENT          a        EMPTY>
<!ATTLIST          a
   b    NMTOKEN  #REQUIRED
   c    CDATA 'y'
   d    CDATA 'z'>
]>
<a b='x' d='1' />
```

Which of the following statements is TRUE about the Element interface in DOM 2?

A. getAttributeNode("b") returns the string "x".
B. getAttribute("c") returns the empty string.
C. getAttribute("c") returns the string "y".
D. An attribute "d" with value "z" immediately appears when removeAttribute("d") is called.

Select two answers.

Explanation: Choice A is not correct because getAttributeNode() returns the Attr node with the specified node name or null if there is no such attribute.

Choice B is not correct because getAttribute() returns the Attr value as a string, or the empty string if that attribute does not have a specified or default value.

Choice C is correct because in this case, the default value of c is y.

Choice D is correct. removeAttribute() removes an attribute by name. If the removed attribute is known to have a default value, an attribute immediately appears containing the default value.

Question 5 (XML Rendering)

Which of the following statements is FALSE about XSL Formatting Objects?

A. The fo:page-sequence formatting object is used to specify how to create a sequence of pages within a document.
B. The fo:table-and-caption flow object is used for formatting a table together with its caption.
C. The fo:table-caption formatting object is used to contain block-level formatting objects containing the caption for the table only when using fo:table-and-caption.
D. The fo:list-item-body formatting object contains the label and the body of an item in a list.

Select one answer.

Explanation: Choice D is the correct answer. The following formatting objects are used to format lists: fo:list-block, fo:list-item, fo:list-item-label, and fo:list-item-body. The fo:list-item formatting object contains the label and the body of the item. The fo:list-item-body formatting object contains only the body of an item in a list. The fo:list-item-label formatting object contains the content of the label of the item.

Conclusion

For each specification covered by the exam objectives, you should put the emphasis on the following:

- The purpose of the specification and its relationship to other XML-related specifications.
- Its benefits when compared to other specifications and how they complement each other in real applications. This will get you ready to correctly answer architecture-related questions, which are often scenario-based.
- The processing model (or rules) of the specification.
- The syntax of the specification.
- The collection of element types and attribute names defined by its namespace and their purpose.

The next article will cover XML Schema and Web Services exam objectives.

JOEL@XMLMENTOR.NET

Premiering
June 2003
at
JavaOne

www.sys-con.com

Millions of Linux Users One Magazine

Linux Business & Technology

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *Linux Business & Technology* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *Linux Business & Technology* will not feature low-level code snippets but will focus instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month will see a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features will include:

- *Advice on Linux Infrastructure*
- *Detailed Software Reviews*
- *Migration Advice*
- *Hardware Advice*
- *CEO Guest Editorials*
- *Recruiting/Certification Advice*
- *Latest News That Matters*
- *Case Studies*

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$49.99

12 ISSUES/YR

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE*

SUBSCRIBE TODAY!

WWW.SYS-CON.COM
OR CALL
1-888-303-5282

LINUX EDGE conference&expo

June 3–5 .....LONDON
June 24–26 .....BERLIN
September ..HONG KONG
October .....CALIFORNIA

FOR ADVERTISING INFORMATION:
CALL 201 802.3020 OR
VISIT WWW.SYS-CON.COM

SYS-CON MEDIA
The World's Leading i-Technology Publisher

Hynes Convention Center: Boston, MA

XML Edge 2003 East

NEWS FROM XML EDGE EAST CONFERENCE & EXPO

▼EAI Consortium Partners with Gartner, Giga, DCI, SYS-CON, and BrainStorm

(Calgary, Alberta, Canada) – The EAI Industry Consortium, global voice for the industry leaders in enterprise application integration, announced its alliance with BrainStorm Group Inc., DCI, Gartner Inc., Giga, and SYS-CON Media in a drive to showcase its member companies' competitive edge in the industry. The partnership with the leading trade show service companies seeks to introduce business process integration within and between organizations using Internet-standard protocols and formats and to inform the marketplace of emerging integration technologies. www.eaiindustry.com

▼Sun Microsystems Launches Sun Developer Network

(Boston) – Sun Microsystems, Inc., has announced the launch of the Sun Developer Network, a new program focused on providing software developers with the content, training, support, and technology access they require to innovate and deliver applications and system solutions faster. The expanded program and portal located at java.sun.com integrates community dialog, content, access to technologies, and advanced learning that will enable Sun and individual developers to more quickly and efficiently implement applications that span multiple technologies, standards, and operating environments. http://sun.com

▼Sun Announces Fully Integrated Web Services Platform, Proposes JCP Specification

(Boston) – Sun Microsystems, Inc., has announced the availability of the Sun ONE Web Services Platform Developer Edition, the industry's first complete and fully integrated platform for Java-based Web services and application development. The Sun Open Net Environment (Sun ONE) Web Services Platform delivers a complete platform offered at a single price and contains all the elements necessary to develop network-based enterprise applications in a single install.

Sun also announced that it has submitted a proposal to the Java Community Process that would expand the way Java developers build integration solutions, by providing a standardized container for business integration components as part of the Java platform. With this new architecture, Sun anticipates that business integration will quickly converge with Web services, helping to accelerate their adoption and reduce integration costs. http://sun.com

▼Digital Evolution Unveils Complete Web Services Management Platform

(Santa Monica, CA) – Digital Evolution, Inc., a provider of Web service management products, has announced its new DE Management Server 2.0. Designed to help businesses conduct secure computing in an open-standards environment, the new server offers a comprehensive management platform that takes the complexity out of managing and using Web services.

The DE Management Server 2.0 comes pre-integrated with existing enterprise security packages and systems, allowing organizations to leverage existing security infrastructure for the SOA rollout. www.digev.com

▼IONA Leads Standards Discussion at Web Services Edge East

(Waltham, MA) – IONA's chief technology officer, Eric Newcomer, led a panel of software industry experts at the Web Services Edge East Conference in Boston. The panel, entitled, "Web Services Architecture: The Next Big Spec, From the Mouths of the W3C Authors" was part of the conference's Web Services track. The panel discussed how the W3C will shape the future of Web services with a successful architecture specification. With Mr. Newcomer on the panel were Heather Kreger, Web services lead architect for emerging technologies at IBM; Michael Champion, lead research and development specialist at Software AG; and David Booth, senior research architect at W3C. www.iona.com

▼Sybase Delivers Technical Session on Achieving Information Liquidity

(Boston) – Robert Breton, senior director of product strategy for the e-Business Division of Sybase, delivered a presentation at the Web Services Edge 2003 East conference, entitled "Achieving Information Liquidity through

Web Services." Sybase defines "Information Liquidity" as the efficiency with which a company transforms data into economic value. Attendees learned how this concept is guiding businesses to gain maximum value from their IT investments through Web services. www.sybase.com

▼World's First SOAP/MIME File Transfer Web Service Released

(Boston) – FileUp Enterprise Edition (FileUpEE), the world's first SOAP/MIME file transfer Web service, was released by SoftArtisans at the Web Services Edge 2003 East – International Web Services Conference & Expo. SoftArtisans also announced the release date for WordWriter for .NET, their newest product for creating Microsoft Word documents on the Web.

With FileUpEE, multigigabyte files, even those in excess of 100GB, are securely transferable within a Web farm environment. FileUpEE scales to the most demanding Web sites, consisting of numerous server farms and multiple levels of security. Content is secured at both the Web server and file server level, ensuring high availability and redundancy. FileUpEE File Servers can be isolated and protected far beyond any other method available today. www.softartisans.com

▼Parasoft Provides Enhanced Automated Error Prevention Tool

(Monrovia, CA) – Parasoft has announced a preview of SOAPtest 2.0, its automated testing tool for Web services. The new version, due out in May, was previewed at the Web Services Edge East show in Boston.

SOAPtest's automated technologies help development teams prevent errors by performing server functional testing, load testing, and client testing with just the click of a button. In addition, developers can also use SOAPtest as a proxy server to view and verify messages between a client and a Web service. www.parasoft.com/soaptest

▼Altova and DataPower Team to Deliver XML Web Services Security

(Beverly, MA and Cambridge, MA) – DataPower Technology, Inc., provider of intelligent XML-aware network infrastructure, and Altova, Inc., provider of XML software tools solutions, have announced the availability of XMLSPY 5 integrated with the Datapower XS40 XML Security Gateway. The unified solution addresses the need for centralized XML Web services security without forcing application developers to alter pre-existing design and deployment practices in any way. www.altova.com, www.datapower.com

▼Actional Unveils Web Services Management Server and Console

(Mountain View, CA) – Actional Corporation has unveiled the Actional Looking Glass Web services management server and console. The new offering enables organizations to reduce the time and cost of managing the impact of change inherent in dynamic Web service networks.

Actional Looking Glass provides a centralized control console combined with a powerful management server that enables users to quickly visualize, understand, monitor, and manage complex Web service networks. www.actional.com

▼Actional Introduces Web Services Initiative, Partners with Microsoft

(Mountain View, CA) – Actional Corporation has announced a detailed initiative to deliver solutions that allow organizations to minimize the impact of constant change inherent in dynamic enterprise Web services environments. As part of this initiative, Actional is announcing a series of new products, strategic partnerships, and customers, demonstrating its market momentum and ongoing commitment to helping organizations realize the full potential of their Web services deployments.

To help deliver on its commitment to helping customers maximize the value of their Web services deployments, Actional is announcing a strategic technology, consulting, marketing, and sales agreement with Microsoft Corporation. The two companies will jointly market and sell their solutions promoting their combined strengths. In addition, James Phillips, senior vice president of marketing and product management at Actional Corporation, has been selected to participate as a member of Microsoft's Infrastructure Advisory Council. www.actional.com, www.microsoft.com

▼Altova's AUTHENTIC 5 Has 200,000 New Users in 2 Weeks

(Beverly, MA) – Altova, Inc., producer of XMLSPY, has announced the widespread adoption and success of their recently released free XML document editor, AUTHENTIC 5. When AUTHENTIC 5 became publicly available under a free license in February, Altova generated approximately 200,000 new users worldwide in 2 weeks.

AUTHENTIC 5 is available immediately for free download at www.altova.com/download_authentic.html and is now offered under a free software license. www.altova.com

Among their users are Agile.Net, CarsDirect.com, the Pocumtuck Valley Memorial Association/Memorial Hall Museum in Old Deerfield, Massachusetts, the University of Regensburg's MedicMed Project, Oxford Analytica, and UC Irvine.

SYS-CON Events Would Like to Thank Our Sponsors and Exhibitors for Making Web Services Edge 2003 East a Success

Actional Corporation
www.actional.com

Actional solutions help organizations avoid the cost and complexity of unmanaged Web services deployments. Actional's Web services management platform provides unmatched visibility, flexibility, and active control across the entire Web service network – ensuring uptime while dramatically reducing the costs of ongoing Web services management.

Altio, Inc.
www.altio.com

Altio offers an XML presentation layer that allows you to bring a live, desktop-style interface to Web applications. Using AltioLive, businesses can integrate their Web applications, legacy applications, and Web services into one unified interface with drag-and-drop capabilities and real-time data feeds.

Altova
www.altova.com

Altova produces and markets XMLSPY 5 Suite, the ultimate Web services development tools suite, featuring a SOAP Debugger and Tester, and many other tools for developing XML Schema, WSDL and UDDI files, and much more.

asp.netPRO
www.aspnetpro.com

This publication targets professional developers who use Microsoft's ASP.NET (Active Server Pages.NET) technology to build Web-enabled applications and business solutions.

ASPstreet.com
www.aspstreet.com

ASPstreet.com is a collaborative portal and marketplace for application service providers (ASPs) and the Web services industry. In the rapidly expanding Web and .NET world, ASPstreet.com is the one-stop hub for all players interested in this emerging marketplace.

Attachmate Corporation
www.attachmate.com

Founded in 1982, Attachmate Corporation, a Microsoft Gold Certified Partner, is a privately held worldwide supplier of mission-critical host access solutions for Fortune 500 and Global 2000 enterprises. Attachmate provides direct, real-time access to back-office systems, Web- and desktop-to-host solutions, plus flexible enterprise application integration and Web services offerings.

Computer Associates
www.ca.com

Computer Associates International, Inc. (CA), delivers software that manages a company's infrastructure by addressing today's most critical business processes, information and technology management challenges. More than two decades of innovation, commitment and quality make CA the technology partner of choice.

Compuware Corporation
www.compuware.com

Compuware Corporation provides business value through software and professional services that optimize productivity and reduce cost across the application life cycle. Pattern-based, model-driven development with Compuware's OptimalJ provides one of the best means to tackle the challenges development organizations face today.

Conquer-IT! (Trans-World Resources, LLC)
www.twresources.com

Think all J2EE training is similar? Trans-World Resources has provided exceptional J2EE training on behalf of BEA, according to Bill Lawson, senior manager, BEA.

DataPower
www.datapower.com

DataPower provides enterprises with an intelligent, XML-aware network infrastructure to ensure unparalleled performance and security of next-generation XML Web services applications. DataPower's patent-pending XG3 technology powers the industry's first XML-aware networking devices to provide immediate return on technology investments while streamlining application deployments.

Digital Evolution
www.digev.com

Tested in the Fortune 500, the DE Management Server 2.0 is the only Web services management application that delivers enterprise-level security, UDDI, multitransport protocols, routing, and performance monitoring for a complex enterprise. Digital Evolution's products and solutions give organizations the power to manage their business processes within a standards-based, vendor-neutral framework.

EAI Industry Consortium
www.eaiindustry.org

The EAI Industry Consortium is a nonprofit global advocacy group developed to promote enterprise application integration through sponsored research, the establishment of standards and guidelines, best practices, and articulation of strategic and measurable benefits. The member-driven consortium, designed as an EAI information hub, encompasses marketplace education, resource tools, and EAI trends, providing members with a venue to develop, create, and debate.

Ektron Inc.
www.ektron.com

Ektron is the vendor of choice for flexible, scalable, and affordable solutions designed to achieve Web content success – today, tomorrow, and beyond. Worldwide, more than 7,000 organizations trust Ektron to help solve their real-world Web content management problems. What's your Web problem?

Forum Systems
www.forumsys.com

To maintain trust with business partners, companies cannot afford to leave entrusted information susceptible to tampering or theft. While most companies focus on XML security in transit, the network is most vulnerable. Forum Systems offers a complete answer, enabling companies to extend their Web services without extending risk. The FS Sentry 2500 ensures integrity through digital signatures, document structure validation, and document translation. It also provides security, rendering encrypted information useless to violators on the network or in transit.

Hewlett-Packard
www.hp.com

HP showcased Web application and services management tools and solutions that enable software developers to easily design and develop for manageability. HP OpenView and Eclipse development tools and new technologies for voice interactive services streamline the development process.

HSPstreet.com
www.hspstreet.com

HSPstreet.com is a collaborative portal enabling buyers to find the appropriate Web hosting plan and partners. In the rapidly expanding Web hosting world, HSPstreet.com is a one-stop hub for all players interested in the HSP marketplace.

IONA
www.iona.com

IONA is the leading provider of Rapid Integration software, with more than 4,500 customers worldwide. IONA's rapid integration software products are built on service-oriented architectures that increase reuse of software assets to deliver lasting results, standards-based software that enables vendor independence, and incremental deployment capabilities that lower the customer's risk.

Itellix Software Solutions
www.itellix.com

Itellix, a software products and services company, focuses on the use of contemporary technologies to realize technology-driven business initiatives. Its flagship product, Wisiba, is a platform-independent, standards-compliant Web services management suite that facilitates organizations to derive commercial value from their Web services initiatives. The Wisiba product suite is composed of Wisiba – Nucleus, Commerce, Optimizer, Intelligence, and Orchestra.

iTKO, Inc.
www.itko.com

iTKO develops some of today's most complex CRM, Web, and back-office applications. iTKO's latest innovation, LISA, made its debut at the Web Services Edge East Conference & Expo and has everyone talking. LISA is a no-code unit, functional, regression, and load-testing product that will change the way you feel about automated testing forever.

Java Developer's Journal
www.sys-con.com/java

Java Developer's Journal is the premier independent, vendor-neutral magazine serving the information needs of the entire community of developers in the Java programming language and Java platform.

JavaWorld
www.javaworld.com

For an audience that demands comprehensive, hands-on information about the news and trends in Java technology, no other information source can match *JavaWorld*'s content. *JavaWorld* is assembled by an award-winning editorial team and authored by seasoned Java developers and industry experts.

Jinfonet Software
www.jinfonet.com

Jinfonet Software is the developer of JReport, a 100% Java reporting tool, written to run on any platform, access any data source, and create any report. JReport Designer is a visual report design interface; JReport Enterprise Server is a high-performance, J2EE-compliant server for deploying reports over the Web.

McCabe & Associates
www.mccabe.com

McCabe & Associates enables IT to deliver better applications by providing products and process that implement a relevant, repeatable, and measurable approach to managing software changes and their effects on the testing and quality of applications. McCabe products include McCabe QA, McCabe Test, and McCabe TRUEchange.

Melissa Data
www.melissadata.com

Melissa Data, founded in 1985, is the leading provider of data-quality solutions to help you achieve the highest level of quality contact information. You'll find our versatile line of software, components, database, and services are easy to use and cost-effective. You'll save money, boost response rates, and increase your bottom line.

Merant
www.merant.com

Merant's PVCS products help you organize, manage, and protect software development assets and improve development efficiency. Leverage the common-use interface of Visual Studio .NET with PVCS to gain greater productivity and control with version/build management, issue and change management, and life-cycle development.

Microsoft Corporation
www.microsoft.com

Introducing Visual Studio .NET – visionary yet practical, the single comprehensive development tool for creat-

ing 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, and 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.

Mindreef, LLC

www.mindreef.com

Mindreef SOAPscope is an easy-to-use, toolkit-independent diagnostic aid for developers, testers, and application support technicians who must isolate Web services problems. SOAPscope has a powerful logger/viewer that shows the SOAP communication flow, making it easy to view, isolate, and debug Web services problems.

My SQL

www.mysql.com

MySQL develops, markets and supports the MySQL database server, the world's most popular open-source database. With an estimated 4 million installations and over 27,000 downloads per day, MySQL is becoming the core of many high-volume, business-critical applications for companies like Yahoo! and Cisco.

.NET Developer's Journal

www.sys-con.com/dotnet

.NET Developer's Journal covers everything of interest to developers working with Microsoft .NET technologies – all from a completely independent and nonbiased perspective.

OASIS

www.oasis-open.org

OASIS is the nonprofit, international consortium that has been providing open solutions for electronic data interchange since 1993. Dedicated from its inception to the technology now known as XML, OASIS is the world's largest independent, vendor-neutral organization for the standardization of XML applications in electronic commerce. The primary mission of OASIS and its members is to identify and resolve interoperability issues that exist between XML applications and technologies.

Oracle Corporation

www.oracle.com

Oracle Corporation 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.

Parasoft Corporation

www.parasoft.com

Parasoft is a leading provider of error-prevention tools that help companies improve their software development processes. These tools assist teams working on C/C++, Java, Web, and enterprise applications to significantly reduce costs by shortening development cycles, improving overall quality, and reducing time to market.

PerfectXML

www.perfectxml.com

The main focus of PerfectXML is XML, Web services, and related technologies for business people and technologists – from both a practitioner and learner perspective. The PerfectXML team works hard to provide best collection of well-organized links, developer-oriented articles and other content, up-to-date news, code samples, and an exhaustive listing of software and tools available.

Rational Software

www.rational.com

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.

Sams Publishing–Pearson Tech Group

www.samspublishing.com/index.asp

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.

SD Times

www.sdtimes.com

SD Times is the newspaper of record for the software development industry. It provides news, news analysis, specialized features, and comprehensive analyses on new products, alliances, and emerging market trends for software and application development managers, IT managers, and ISVs, who manage development projects. Subscriptions are free.

SlickEdit, Inc.

www.slickedit.com

SlickEdit Inc., provides software developers with the most comprehensive and flexible code editor available. Visual SlickEdit, proven across a wide range of programming languages and on Windows, Linux, Unix, and zSeries mainframe platforms, enables even the most accomplished developers to code faster and meet increasingly aggressive deadlines.

Softartisans, Inc.

www.softartisans.com

With enterprise-class products like ExcelWriter and FileUp, SoftArtisans assists clients on any development platform to build robust reporting, file transfer, and Web-based solutions. Over 16,000 customers in more than 70 countries benefit from SoftArtisans products and technical services.

Software AG

www.softwareag.com

Software AG, Inc., is a pioneer in XML solutions and a leading global provider of system software and services enabling enterprise data integration and management. Our products and solutions focus on standards-based XML integration such as Web services and enterprise content management.

SpiritSoft

www.spiritsoft.com

SpiritSoft, the leading provider of integration software using JMS and JCache technologies, enables developers to align IT resources on a unified foundation built on open standards, which lowers the cost of an existing IT infrastructure and boosts performance and return on investment. SpiritWave Message Server, the leading Java Message Service (JMS) implementation, provides reliable, flexible, and secure messaging to enable flexible integration between new and existing enterprise applications. SpiritWave Open JMS Framework also allows developers to integrate proprietary middleware and offers a range of interface/language bindings for legacy enterprise applications.

Sun Microsystems

www.sun.com

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 for the electronic marketplace.

Sybase, Inc.

www.sybase.com

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

Teamstudio, Inc.

www.teamstudio.com

Founded in 1996, Teamstudio develops and markets award-winning, agile software tools that enhance developer productivity and improve application quality. Product lines include solutions for Lotus Notes, Domino, and Java developers, and Web-to-host integration.

Trilog Group, Inc.

www.triloggroup.com

Trilog Group is the only software company that provides a fully integrated platform for J2EE RAD, BPM, and Web services integration. Fortune 500 companies have used FlowBuilder Visual XSP Studio to drastically reduce development and integration costs by capitalizing on its ultra-rapid, highly visual, XML-centric method to manufacture and assemble enterprise application components and Web services.

Vultus, Inc.

www.vultus.com

Vultus speeds the adoption of best-of-breed Web applications that greatly enhance corporate Internet strategies. Our technology provides a flexible, secure platform to facilitate closer business relationships with new and existing customers. Vultus products are designed to extend long-term IT and business ROI for our customers by adhering to the latest industry protocols.

Web Services Journal

www.sys-con.com/webservices

Web Services Journal is the premier publication addressing the technical and strategic depth of Web services. It is for anyone who wishes to apply the new model for creating and using distributed applications across the Internet, utilizing common interfaces for efficient communication and high-level interoperability.

WowGao.com

www.wowgao.com

WowGao.com is an international leader in Web services deployment, hosting, and portal. GAO Research Inc. (www.GaoResearch.com), offers solutions for modem (ADSL, V.92, V.90,. etc), fax, modem/fax relays, telephony, speech, VoIP, and gateways. GAO Web Services Inc. (www.GaoWebServices.com), specializes in Web services. Its powerful, modular, and scalable J2EE-based UDDI is available for license.

XML-Journal

www.sys-con.com/xml

XML-Journal is the world's leading print and online resource for Internet technology professionals involved with the worldwide development and implementation of XML. Each issue contains the latest news concerning enterprise application integration and Web services, XML standards, new developments in e-commerce, product reviews, tutorials, case studies, and interviews with *i*-technology leaders.

Xtremesoft, Inc.

www.xtremesoft.com

Xtremesoft is the leading provider of software solutions that maximize the availability of applications on the Microsoft platform. These solutions enable businesses to transform and process data derived from their applications into business intelligence upon which decisions can be made.

WS-I and JCP: Creating Value for Enterprises

I'm frequently asked about the difference between portability and interoperability, and am often surprised at how many people refer to one when they mean the other.

WRITTEN BY
GLEN MARTIN

On the surface, the terms are pretty understandable: interoperability means that different systems will work together. Portability means that systems will work in different places. It's clear that enterprise customers need both. How many times have you heard an IT person say, "Our systems don't need to talk to each other?" or "Our deployment needs are never going to change?" (No doubt such folks still have 640KB PCs on their desks.)

But many developers overlook technology advancements and pretend they can go on deploying their applications on the same systems. This despite 64-bit processors waiting in the wings, whose architectures are rather different from those we use today.

Portability doesn't just mean being able to run on another system that you have now. It also means being able to run on another system that hasn't been built yet.

Some developers imagine their applications won't be around that long. COBOL programmers thought this way in the 1970s, but their applications show little sign of disappearing. In fact the opposite is true: there is good money to be made in maintaining COBOL applications today. (Perhaps this was job protection.)

> "The main benefit of the WS-I is the exclusive focus on interoperability, precisely because it's hard to get right"

But I think we prefer to add value by writing new applications, not constantly hacking on old applications to make them run on a new system. So how do we best achieve both portability and interoperability?

Sun was recently elected to the board of the WS-I (www.ws-i.org), an organization chartered with the development of Profiles that chart a course through the maze of XML and Web services standards to create a basis for cross-platform and cross-language interoperability. This is good – both the WS-I itself and Sun's election to the board.

The main benefit of the WS-I is the exclusive focus on interoperability, precisely because it's hard to get right. People are trying to do a lot of different things with the Web. It's easy to focus on just a few uses and determine the requirements that enable those to work. What's hard is figuring out what is needed for the whole breadth of requirements.

I truly hope that the WS-I is up to the task of charting that course through the many and sometimes contradictory standards, and that it will not get bogged down in politics.

Some organizations seem to have more trouble with the smoke-filled room than others. I remember the glorious failure of the ODMG (Object Data Management Group) standard. There was a completely incompatible operating mode written up in an Appendix of the spec to validate it, even though only one leading vendor would ever implement it.

The WS-I has been showing signs of rising above the smoke to do the Right Thing, with the draft of the Basic Profile requiring support for document-based Web services instead of only the overly simplistic RPC method. Document transfer is much more useful for real business-oriented Web services, even if some vendors have strong product focus on RPC.

Sun's inclusion on the WS-I board is good in part because so many developers use the Java platform for creating Web services, and also because Java has successfully achieved a high degree of interoperability through standards.

In his role as lead architect for the Java 2 Platform, Enterprise Edition (J2EE), Mark Hapner, Sun's representative to the WS-I board, has done a great job of ensuring that J2EE applications may be deployed across multiple vendors' solutions with fully secure and transactional interoperability.

But before we lift a celebratory beer at having solved interoperability, let's remember that this is only half the problem. Portability is where the Java 2 platform comes in.

An analogy may be drawn to writing in compiled languages instead of assembly language. Rewriting only the compiler can improve performance of an application, or allow that application to run on new processor architectures. It costs less to rewrite the compiler than to rewrite all our applications.

The Java platform extends this analogy to the rest of the system. The virtual machine may be enhanced to squeeze more performance out of existing hardware, or rewritten to take advantage of evolutionary or revolutionary system changes. The Java APIs that intermediate between the application and external services (including Web services) may be rewritten to reflect the evolution of those protocols. All without changing the application itself.

The Java Community Process (www.jcp.org) is how we standardize Java APIs and the required tests that ensure different implementations exhibit the same behaviors behind those APIs. The Java platform and APIs are developed and maintained by expert groups consisting of architects, senior developers, and technical visionaries from more than 450 companies. Specs and tests are required to achieve application longevity in a changing world.

The WS-I and JCP can together create a good deal of value. The WS-I defines the profiles that enable interoperability, and the JCP abstracts the protocols and services specified in the profiles within APIs. This allows the evolution of infrastructure without rewriting all the applications that utilize it.

GLEN.MARTIN@SUN.COM

AUTHOR BIO

Glen Martin is a J2EE strategist at Sun Microsystems. He leads the marketing and product management team responsible for Java Web Services and Java 2 Platform, Enterprise Edition (J2EE). Glen participated in the Enterprise JavaBeans expert group, and wrote the J2EE 1.3 requirements document and J2EE 1.4 concept document. He has 14 years of broad industry experience in technical and marketing roles, developing products ranging from packet switches to development tools and several points in between.

Quest Software, Inc.

http://java.quest.com/jcsc/ws

Altova

http://xmlj.altova.com/wsdl