SYS-CON MEDIA

JAVA DEVELOPER'S JOURNAL
JDJ

SPECIAL REPORT:
SCOTT McNEALY
2004 PREDICTIONS

PLUS...

Viewpoint: Deutsch's Fallacies, 10 Years After
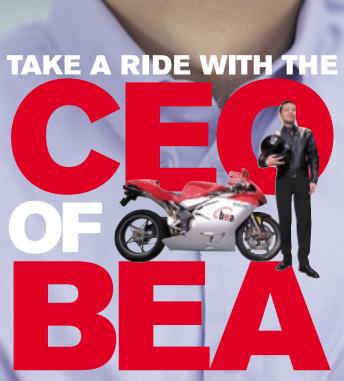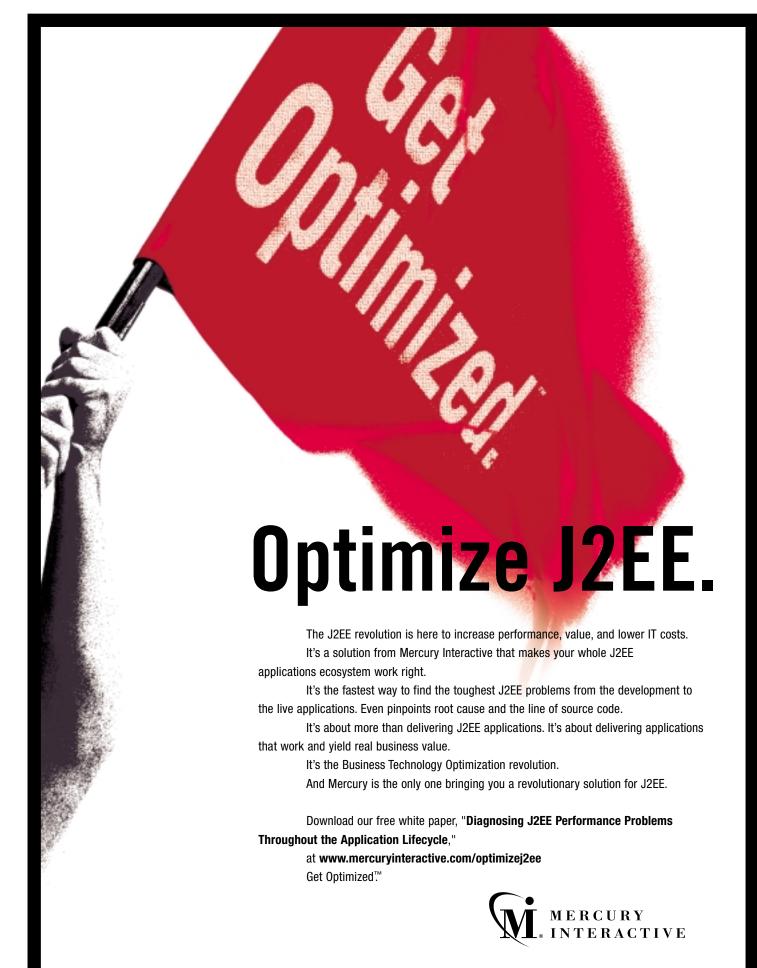
Feature: Multiplatform Frames

Data Integration: Which EII Solution Is Right for You?

TAKE A RIDE WITH THE
CEO OF BEA

EDGE 2004 EAST
International Conference & Expo
BOSTON, MA
FEBRUARY 24-26, 2004
see pages 47-60 for details

JDJ contents

Cover Story: Exclusive Interview

JDJ Interviews...

Alfred Chuang

Founder, President, and CEO of BEA

Interviewed by Alan Williamson

16

Special Feature

24

2004 PREDICTIONS

Where Is *i*-Technology Going in 2004?

Compiled by Jeremy Geelan

Features

32

Core Object Principles in OO Programming
by York Davis

42

Multiplatform Frames
by Todd Brett

Welcome to *JDJ,* **Again...**

Fuat Kircaali

With this January issue, **JDJ** is entering its ninth successful year of publication, and we have achieved this success by serving the most influential readers – like yourself – around the globe.

For the past eight years, *JDJ* has become the single most popular *i*-technology magazine in the world, its circulation exceeding all other Internet/information technology developer magazines published anywhere. In fact, its only competition may be other **SYS-CON** titles, such as *LinuxWorld Magazine*, *Web Services Journal*, *.NET Developer's Journal*, and *MX Developer's Journal*. Again, the greatest reason for our success is you, and your daily support and feedback.

In this issue, you'll find a great collection of articles and columns. We have several new editorial offerings (as you may have noted from our redesigned cover). Long-time *JDJ* editor Alan Williamson has an exclusive interview with the founder and CEO of BEA Systems, Alfred Chuang, which you'll find very interesting. One of the photos of Chuang that accompanies this interview was taken during the last BEA User Conference in Florida, as he was entering the stage to deliver his speech. I was present at that conference and it was a memorable moment. I wonder if Alfred and another famous biker,

Kertzman reading the premier issue of *PBDJ* in 1994

Alfred Chuang
CEO of BEA

Jocelyne Attal
ex-IBM WebSphere Chief

ex-IBM WebSphere chief Jocelyne Attal, attend the same "American Chopper" shows…

Jeremy Geelan's compilation of 2004 predictions from *i*-technology experts is another exclusive we wanted to share with you. Among the names you'll see in this story are Mitchell Kertzman, whose connection with SYS-CON goes way back. When we announced our first technology magazine, *PowerBuilder Developer's Journal*, in 1994 on

Fuat Kircaali is the founder and CEO of SYS-CON Media.

fuat@sys-con.com

CompuServe forums, we received instant and overwhelming interest from the community. I remember one e-mail from Boston offering assistance. When I replied to it and asked who he was and how he could help us (remember the old CompuServe numeric e-mail addresses?), his response was, "My name is Mitchell Kertzman and I'm the CEO of Powersoft. How can we help you with the magazine?" You could say SYS-CON was born with that e-mail.

We asked other tech leaders for their opinions for this month's cover story. Larry Ellison declined to comment at this time, perhaps until a decision is reached from the Department of Justice on the PeopleSoft deal. We'll check back with Oracle and Ellison in the future for an exclusive story. An interview with Ed Zander, discussing his recent appointment to the CEO post of Motorola, would also make a good read. These are some of the topics we'll be exploring for our upcoming issues.

I recently met Scott McNealy at last year's Comdex show in Las Vegas. You may be thinking, "Isn't it a little late for a first time meeting, after eight years of publishing *JDJ* :-)?" Well, maybe, but he was kind enough to take the time to answer our questions days before this issue went to press and we thank him for that. And we have been working very closely with a number of Sun executives over the past eight years, including Alan Baratz, George Paolini, and James Gosling. I will never forget Gosling's opening keynote at our Web Services Edge Conference in New York City, two weeks after 9/11. His first words were, "My keynote appearance here was originally scheduled as part of a sequence of events in places as far away as Vienna and Stockholm. This is the only one that was not cancelled." It was truly a historical event and a message from New York to the world.

• • •

For the past 10 years, we have been working hard to bring you the best magazines in the industry, and you will see a number of new offerings from SYS-CON in 2004. Happy and healthy New Year wishes!

Deutsch's Fallacies,
10 Years After

Ingrid Van Den Hoogen

In the fall of 1991, when mobile computing involved a hand truck and an extension cord, the idea of an everything-connected world was a leap of faith to some and a really crazy idea to most. But Sun's engineers were already working on notebook computers, and Peter Deutsch, one of Sun's original "Fellows," was heading up a task force to advise Sun on its mobile strategy.

Deutsch just called 'em like he saw 'em. When he got to Sun he began to consider some of the existing notions some engineers had about network computing, some of which were downright foolish.

Coming off a stint as chief scientist at Park Place Systems, Deutsch was looking to hang out and cogitate with and bask in Sun's intense engineering culture. He was a key designer and implementer of the Interlisp-D system and a significant contributor to the design of the Cedar Mesa language and the Smalltalk-80 programming environment. But he hadn't gotten into networking. It might have been some sort of intellectual hazing ritual that made Deutsch co-chair of a mobile computing task force. Or it might have been brilliance.

Bill Joy and Dave Lyon had already formulated a list of flawed assumptions about distributed computing that were guaranteed to cause problems down the road: the network is reliable; latency is zero; bandwidth is infinite; and the network is secure. James Gosling, Sun Fellow, Chief Technology Officer for Java Developer Platforms, and inventor of Java, had actually codified these four, calling them "The Fallacies of Networked Computing."

"It's a sort of funny thing," he says, "that in the large-scale world, networking didn't really exist in 1995. But since Sun was founded in 1982, networking has been at the core of what we do. We cut ourselves on all these problems pretty early on."

What Deutsch saw with fresh eyes was that, despite Gosling's warning, as engineers – inside and outside Sun – designed and built network infrastructure, they kept making the same mistakes, based largely on the same basic yet false assumptions about the nature of the network.

"The more I looked around at networking inside and outside Sun," Deutsch says, "the more I thought I could see instances where making these assumptions got people into trouble." For example, Deutsch could see Gosling's Fallacies coming into play as Sun moved its operations from downtown to its glamorous new campus near the San Francisco Bay.

"There was a lot of thrashing around about the topology of the network for the corporate intranet, where routers should be, etc.," Deutsch recalls. "Things broke all the time. My recollection is that it was watching all that thrashing around that led me to numbers five and six – that there's a single administrator and that the topology won't change." Number seven, that transport cost is zero, coalesced as Sun discussed creating a wide area network to connect the Mountain View campus with a new lab on the East coast.

When Deutsch wrote the list, by this time seven items strong, into a slide presentation, "It was no big deal," he says. Neither was there a roar of acclamation. Rob Gingell, Sun vice president, chief engineer, and Fellow, remembers it as an mmm-hmm moment rather than an ah-ha moment.

But putting it down on paper, codifying it, made all the difference. "The list Peter wrote down was a very pithy summary of the pitfalls people typically fall into," Gosling says. "We had many conversations about it. There would be a presentation where somebody would be proposing some design, and somebody would point out, 'You know, that kind of depends on the network being reliable. And that's false.'"

Gosling added the Eighth Fallacy – the assumption that the network is homogeneous – in 1997 or so. "It reflected a mismatch between our perception and others' perceptions of the network. That the network is homogeneous is never a mistake we've made. But it was clear that lots of people on the outside had a tendency to fall into this."

Ingrid Van Den Hoogen is the senior director of software strategy and marketing for Sun Microsystems. She was the founder of the JXTA project at Sun, and a member of the Jini team.

ingrid.vandenhoogen@sun.com

DESKTOP   CORE   ENTERPRISE   HOME

Get the complete picture

Features, Performance and Control

Discover the ILOG JViews Graphics Components

You're developing a sophisticated user interface for a desktop, applet or servlet application – it needs to provide displays that go far beyond what Swing and HTML offer. How can you be sure it will have the features, performance, customization and scalability to enable your end-users to make better more informed decisions, faster?

With ILOG JViews, you get comprehensive graphical libraries & tools, resources, and maintenance services so you can focus on the implementation, confidently completing your application in less time and at less cost.

Quickly and easily build:
- Gantt and resource displays
- Graph layouts, diagrams, workflows
- Geographic map displays
- Realtime data charts
- Custom monitoring and control screens
- Network and equipment management screens

Get a JViews Info Kit – Learn more, test drive an Eval.
Go to: jviews-info-kit.ilog.com or Call: 1-800-for-ILOG

ILOG®

Changing the rules of business™

Joseph Ottinger
Java Enterprise Editor

DESKTOP

CORE

ENTERPRISE

HOME

Let's Do **Better**

There's a J2EE tie-in. I promise.

A fellow user mentioned something the other day about using libraries in Java. He said, and I paraphrase, that he simply didn't feel comfortable using a library if it couldn't be explained in one page. For the sake of argument, I'll assume he didn't mean one 400K page.

I'm not sure I wholly agree with his statement, but his point is well taken. When I look for a tool, I want that tool to have a very clear purpose and documentation. A name should be an easily associated mnemonic, perhaps a direct tie-in to purpose, and the documentation should allow the user to quickly get a sense of not only what the product does, but how and why, too.

Some examples of good product names, in my opinion: EXML, RSSLibJ, commons-httpclient. Some bad ones: commons-digester, maven, blissed. I'm not saying the ones named well are great products; I have my reservations about them, even the one I co-authored. Nor am I saying the products whose names I dislike are not good products: I'm simply saying that their names don't help you associate the products with what they're for.

I think virtually everyone understands the problem of good documentation: you'd be hard-pressed to find any programmer who hasn't gone to some project's documentation, read for a while, and come away scratching her head with no clearer picture than she had before. Maybe that's happened with popular projects – I know that some of my own general-purpose projects are still "blessed" with obscure docs.

I'm hardly exempt from the problem of documentation. I'm a fairly abstract thinker in a lot of ways, and a "big thinker," so I have a tendency to solve problems in abstract ways, making things harder to explain. It's happened a number of times in my career that I've intuitively architected a solution and been unable to explain it to the teams around me…until later, when the problems my solution addressed manifested themselves. (True story: one team refused to take my solution as anything more than the rantings of a mad hatter, and chose to use a simpler, more straightforward architecture. Months later, when the localization problem I was trying to solve became apparent, they – and I – realized that I'd been on the right track all along, and we ended up going back to the plan I'd suggested first. I've never had a deserved reputation as a great communicator.)

My difficulties are directly related to my mode of thought because I tend to be a bit scattered and draw technical material from a lot of sources related to the humanities. I can shrug and say, with a light sprinkling of seriousness, that I'm an artiste and can't be held to reasonable expectations. That said, I don't see that same scattershot approach in every project, so why do so many of them have poor or unclear documentation? We're nominally engineers; why can't we approach our solutions from an engineer's cut-and-dry viewpoint and say, "Here's the problem; here's a solution; here's why this solution was chosen; here's how to make it go"?

I don't have a clear answer. If I did, I'm sure my own wider-scoped projects would have better documentation. I can just use that programmer's statement about a one-page document as a guide limning my path, and as a measuring stick that I can hold my own documents up to. Is the document as short as it can be? Does the document have a premise? Is it directly answering that premise? If I was reading it without foreknowledge of the subject matter, would I be able to understand what the document was trying to say? Would it contain useful references to other documents to give me the bits outside of the basic premise?

I know that I sound like a professor of language, asking those who write to approach the task as if it were a craft. I imagine many are shuddering with bad memories of college essays, thinking you'd gotten away from crafting term papers and theses. Well, you have – I don't ask anyone to write a doctorate-level paper when all you are trying to do is tell someone how to write a Webwork interceptor. However, the basic skills still apply – who, what, when, where, why, and how. Without these being at the very least addressed and discarded, a document will always sound vaguely unfocused, and that's torture for your readers.

By now, you're reading this wondering why a Java Enterprise editor is lecturing on basic documentation skills. I promised a tie-in, and here it is: recently, Tod Nielsen (executive VP of BEA) wrote a missive I disagreed with about the future of J2EE being predicated on clarity and simplicity. I saw his point, but I think he's reacting to the lack of effective writing on the industry's part. We write complex systems and don't bother to explain them in simple ways, so we're left perceiving their complexity, and people who aren't interested in complexity for complexity's sake just don't care. Donald Knuth imagined literate programs as programs that you would read, sitting by a fire with a glass of something enjoyable handy. If architectures were designed with this kind of perusal in mind, think of the future time spent maintaining them: it'd be a joy, comparatively, and Mr. Nielsen's plea for a simpler API would sit unneeded.
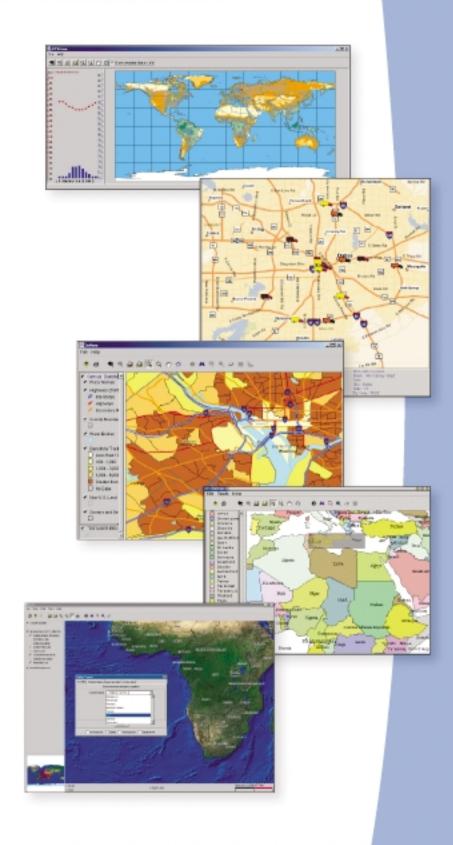
We need to work on our communications, all of them. Our systems, regardless of their underpinnings, need to be explained in simple, clear terms. Anything less dooms us to the amateurism of which our industrial competitors accuse us.

Let's do better. ✎

Joseph Ottinger is a consultant with Fusion Alliance (www.fusionalliance.com) and is a frequent contributor to open source projects in a number of capacities. Joe is also the acting chairman of the *JDJ* Editorial Advisory Board.

josephottinger@sys-con.com

MapObjects—Java Edition

Introducing a Pure Java Solution for Adding Dynamic Maps to Your Applications

Imagine the possibilities of building custom applications that allow users to see information visually represented on a map. With such a tool, your users could spot patterns and trends in data that might otherwise go unnoticed, resulting in better, more informed decision making.

The ESRI® MapObjects®—Java Edition software is a suite of more than 900 Java developer components that you can use to build custom, high-powered geographic information system (GIS) applications or applets at the client, Web, and server tiers.

With MapObjects—Java Edition, you can

▸ Build applications with a wide range of mapping and GIS capabilities (labeling, panning, zooming, measuring distances, querying data, and much more).

▸ Easily drag and drop visual JavaBeans into an integrated development environment at design time.

▸ Use Swing Media components to quickly create a user-friendly interface.

▸ Combine multiple data sources.

▸ Display a variety of data types.

▸ Deploy J2EE server-based applications.

MapObjects—Java Edition

Mapping and GIS Tools for the Java Development Environment

1-888-332-2320

www.esri.com/mapobjectsjava
info@esri.com

Apache **James**

A customizable e-mail server solution

by Kulvir Singh Bhogal
and Kwang Sik Kang

I n today's society, it's second nature to contact family, friends, and colleagues via e-mail. E-mail has become so integral to our daily ritual that we cannot stay away from it as we make use of personal digital assistants (PDAs) and other pervasive devices to have it readily available. How do these billions of digital messages get to users across the globe?

A client/server setup is required to participate in the exchange of e-mails. It was just a few years ago that expensive server software packages were used for the typical e-mail solution. Cheaper solutions have been plagued by a lack of robustness and scalability. To the avail of shrinking corporate budgets comes the Java Apache Mail Enterprise Server (known as Apache James).

Apache James is a free, open source, robust offering from the Apache Software Foundation that aims to provide enterprises with a Java-based mail server solution. James provides a Java API for developers to process e-mail services. In this article, we'll explore James and what it can offer your enterprise operation.

Understanding E-Mail

In the world of e-mail, the senders and recipients of messages act as clients to e-mail servers. Senders create an e-mail using a Mail User Agent (MUA) and transmit the message to an e-mail server using the Simple Mail Transfer Protocol (SMTP). The e-mail server uses the same protocol to contact the recipient's e-mail server. Recipients then use the Post Office Protocol (POP3) or alternatively the Internet Message Access Protocol (IMAP) to retrieve the message from their e-mail server. *Note:* IMAP support is currently experimental in James 2.1.3.

SMTP

As mentioned, e-mail servers use SMTP to transmit messages across the world. The default TCP port for this protocol is 25. Accordingly, network administrators need to configure firewalls for access to the SMTP port.

POP3

POP3 is used to retrieve messages from the e-mail server. The default TCP port for this protocol is 110. Clients must provide security information used for authentication with the e-mail server. Upon successful authentication, clients can access their messages from their respective mailboxes.

Getting Hold of James

The software is available for download from the Apache James homepage at http://james.apache.org/. James was in version 2.1.3 at the time of this writing; this version requires Java 2 (Java Runtime Environment version 1.3 or 1.4 as of 2.0a3).

Firing Up James

Before you begin, ensure that the system environment variable, JAVA_HOME, points to the JRE 1.3+ home directory. To fire up James, simply navigate to the bin directory created upon extracting the contents of the downloaded archive file. In this directory, locate and execute the "run.bat" script. Upon running the script, a confirmation will be displayed that James is listening on ports 25 (SMTP) and 110 (POP3) (see Figure 1).

The configuration files are unpacked the first time James runs. At this point stop the server, make the necessary configuration changes, and restart the server for the modifications to take effect. A needed modification is in the config.xml file located in the <james-installdir>\apps\james\ SAR-INF directory.

```
<servernames autodetect="true" autodetectIP="true">
<!-- CONFIRM? -->
<servername>localhost</servername>
</servernames>
```

Add the Domain Name Service (DNS) namespace you want served by James. For example, to have James provide e-mail services for somedomain.com, the configuration would look like the following:

```
<servernames autodetect="true" autodetectIP="true">
<!-- CONFIRM? -->
<servername>localhost</servername>
<servername>somedomain.com</servername>
</servernames>
```

Note: The assumption is that the necessary upstream configurations are in place for the DNS and Mail Exchanger (MX) records.

Administering James

Remote administration to the James server is available through Telnet. The "Remote Manager Service" listens on port 4555 by default. Access this service using the following command:

```
telnet localhost 4555
```

Replace "localhost" in the above code with the appropriate host name or IP address. The default login/password for the Remote Manager Service is root/root. To alter this information, edit the config.xml file located in the <james-installdir>\apps\ james\SAR-INF directory. Look for the following section:

```
<administrator_accounts>
<!-- CHECKME! -->
<!-- Change the default login/password. -->
<account login="root" password="root"/>
</administrator_accounts>
```

Kulvir Singh Bhogal works as an IBM e-business consultant, devising and implementing solutions at customer sites across the nation.

kbhogal@us.ibm.com

Kwang Sik Kang is a consultant with IBM Software Services for WebSphere.

kkwang@us.ibm.com

The Remote Manager Service provides the user with an easily accessible help facility through the command HELP.

Now, begin administration of the e-mail server by adding user mailboxes. This is accomplished by issuing the following command:

```
adduser [username] [password]
```

For this exercise, we added two users, kulvir and kwang, with the same password of security.

```
adduser kulvir security
adduser kwang security
```

As shown in Figure 2, James will not echo the process of you creating the user.

Sending and Receiving Some Mail

At this point, e-mail clients such as Lotus Notes and Microsoft Outlook should be able send and receive messages from the e-mail accounts just created. Consult your product documentation regarding proper account configuration for your client of choice.

The Mailet API

Similar to Java servlets running on an application server, the Apache Software Foundation allows James to act as a container that can run mailets that extend the capabilities of the mail server. Mailets are Java applications, developed using the

Figure 1 Successful startup

Figure 2 Adding users

Figure 3 Receipt of e-mail

mailet API, that perform various mail-processing tasks including auto-response engines, spam filters, and e-mail indexing. The mailet API, along with sample mailets, is bundled with the James distribution.

For our exercise, we will enable "AddFooter", a prepackaged mailet. First, modify the "config.xml" file by adding the code in Listing 1. Let's discuss two important attributes of the mailet tag. The class attribute refers to the mailet's Java class. The match attribute refers to the matcher, a Java class that implements the Matcher API. Matchers return a collection of recipients that satisfy the defined conditions. There are a number of matchers prebundled with the James distribution. The "All" matcher returns all recipients for all messages processed by a mailet.

James registers mailets during initial startup. Restart the server to enable the newly defined mailet. Send an e-mail to one of your registered accounts. The recipient should see the specified footer text added to the end of a message sent to him or her (see Figure 3).

Extensibility

The power of James lies in its ability to extend its capabilities through custom mailets and matchers. Using the APIs provided by the Apache Software Foundation, you can create your own setups to process mail, satisfying virtually any business requirement required of a mail server. The James Web site has a good amount of documentation about how to go about creating your own mailets and matchers.

Mailing Lists

James also supports mailing lists that allow individuals to subscribe/unsubscribe to topics. It's simple to configure and use. For more information regarding the James mailing list feature, view the James FAQ documentation.

Contributing to James

Like other Apache open source projects, James relies on the programming community to contribute code enhancements to yield a more powerful and stable application. It's this collaborative effort that has led to what we see in James today. You can learn more about contributing to James at http://james.apache.org/contribute.html.

Conclusion

As this article illustrated, James is simple to configure and run. Coupled with the power to design customer matchers and mailets, it represents a highly user-friendly, extensible e-mail server solution, customizable to fit your enterprise needs. 🖊

Listing 1
```
<!--
AddFooter appends the specified text to each message
place this code in the processor name="transport"
section -->
<mailet match="All" class="AddFooter">
 <text>
   ----------------------
Somerandom Sender
XYZ Corp.
(999)999-9999 [Office]
(888)888-8888 [Cell]
"Peace cannot be kept by force; it can only be
obtained by understanding." – Albert Einstein
 </text>
</mailet>
```

EXCLUSIVE:
JDJ interviews...

Alfred Chuang

Founder, President, and CEO of BEA

Interviewed by Alan Williamson

*First of all let me thank you, Alfred, for taking the time to talk to us at **JDJ**...*

JDJ: Let's go back to that day in October 2001, when you stepped into BEA as its new CEO. What was your perception of the Java industry at that time?

AC: I was tremendously excited by the capabilities of J2EE, and the huge acceptance it had already achieved in the enterprise. For the first time in enterprise software there was a realistic hope that a standard software platform for enterprise computing could be adopted. J2EE had already proven its suitability for mission-critical applications, and competitors like IBM, Sun, Oracle, and BEA were able to put aside their differences enough to make J2EE a viable and powerful standard. I was also very proud of BEA's leadership role in driving J2EE forward.

As much as I was excited about the potential of enterprise Java, I was well aware that the platform was still too difficult for the broader community of developers to adopt en masse. Changing that became one of our top priorities.

JDJ: Nearly two years on, how has your view of the Java industry changed?

AC: The Java standard has continued to move forward, but at too slow a pace. We're committed to the Java Community Process, but we're unhappy that it takes as long as it does to get innovations added to the standard.

Look, we have a huge lead on .NET, as far as what is possible to put into production for a mission-critical application at a customer site today. They are years away from catching up. But the community takes that too much for granted. Microsoft's powerful marketing machine can convince many people that they are catching up or even passing J2EE. So a one- or two-year lead isn't enough. We cannot be complacent.

This is why we work so hard to drive our innovations into standards. There are many customer problems yet to be solved. Customers don't want proprietary solutions, but they do want their problems solved now. We have no choice but to innovate fast, give those innovations to our customers, and simultaneously drive our innovations into the standards.

I also think that the Java camp wastes too much time attacking Microsoft. In the real world, customers are going to have both Java and .NET in their organizations. They have to make these technologies work together. They aren't interested in hearing anyone trash the competition. They are much more interested in hearing how we'll be able to work well in a heterogeneous environment. So we've taken great care to make sure we have great interoperability with Microsoft. In many ways, customers realize that WebLogic's proven mission-critical capabilities and openness make it the ideal on-ramp to the enterprise for the departmental .NET applications that they have.

TAKE A RIDE WITH THE CEO OF BEA

JDJ: Outside of Sun and IBM, BEA is seen as the single biggest Java player in the industry. You run a billion-dollar company. This is quite a success story for a pure Java company, yes? Where do you see the future revenue of BEA coming from?

AC: Yes, I agree – it's a rare $1 billion company that has been able to go toe-to-toe with the IBMs of the world and win. We're proud of that. It's a success story made possible by BEA's ongoing ability to identify trends well in advance, deliver the best technology, and work well with a broad expanse of great partners. Simplistically put, yes. But over the years, that formula, well executed, has brought us this success.

We are using that same formula as we look to drive future revenue. Specifically, we identified the need for and market shift toward application platform suites over two years ago – and we've delivered a product in line with that vision, WebLogic Enterprise Platform 8.1. Each of the components of the platform are best of breed in their categories – application server, integration, portal, security, etc. – and the credibility we get from that is essential. We see our business growing as we continue to gain market share in these individual markets but, more important, as the platform as a whole continues to gain acceptance, every quarter we see the percentage of customers who are buying the whole platform go up. Customers really understand the value of a service-oriented architecture and the notion of an application platform suite.

JDJ: Why does a company buy from BEA? Are they looking for a Java application server, or are they looking to you for guidance on building and deploying their enterprise applications?

AC: Enterprise customers never just buy a product, whether it's an application server or anything else. They are looking for a partner they can trust.

They buy BEA because they trust our products – BEA technology has proven itself across 15,000+ customers, in some of the most demanding IT environments.

They buy BEA because they trust our independence. We aren't tied to any one operating system, database, hardware platform, etc. We work well with all of them, because we have to. Customers trust us because we don't have any agenda other than application infrastructure.

Finally, they buy BEA because they trust our ability to help them simplify their IT environment. They know that our technology is integrated and interoperable. We don't introduce unnecessary complexity into IT, unlike our competition. We've done the hard work of integrating and converging a complete application platform suite. Our competitor makes the customer responsible for that work, and counts on that complexity to sell services.

Our competitor sends out 54 CDs for what we deliver on just one, even though we provide more functionality. That's because they are integrated in brand only, and their "platform" is an assemblage of hundreds of disparate products acquired over the years and shellacked with the same veneer.

Complexity is contagious. A customer inherits its vendor's complexity; and complexity is one of a customer's biggest pain points right now. In response, we've made it simpler for customers to develop and deploy by giving them an integrated product, an easier-to-use tool in WebLogic Workshop and standards-based technology. Enterprise software is not going to be easy any time soon, but it can be a lot easier than it is today.

"Sun has already ceded thought leadership on enterprise Java to BEA and others"

JDJ: When your sales force is in the field, are they still fielding questions regarding what/why Java? Is the awareness of Java's ability in non-IT verticals well known?

AC: We don't get many questions about "why Java" today. The strength of the platform is well known. It is *the* enterprise standard at this point – that's true for all industries we sell into. To that point, we're in 100% of all global Fortune 500 financial securities, telecommunications, health care, pharmaceuticals, and network communications companies. So Java is pretty well known.

Customers were looking for standards-based technology, because their environments were so heterogeneous – they found that in Java.

JDJ: To date, what do you see as Java's single greatest success?

AC: Without a doubt, it is offering IT their first ray of hope that there will be a standard software platform for the enterprise. Java provides unity in a truly cross-platform environment. No other enterprise standard delivers that.

Imagine how hard it is to get powerful competitors like IBM, Sun, and BEA to work well together. That's a testimony to the promise of enterprise Java.

JDJ: Let me give you the keys to Doc Brown's DeLorean time machine so you can take a trip back in time. Which event in Java's (or BEA's) history would you most like to change?

AC: The mistake I would change is that too much time and energy has been wasted obsessing about Java on the desktop. Look, if we do the right things on the enterprise platform, the desktop almost doesn't matter. Ironically, it's not Java on the desktop that is going to keep Microsoft from owning all computing, it's Java on the server. The better enterprise Java gets, the more powerful applications can be, and there's nothing Microsoft can do to get in the way of that. Fighting them on the desktop is not going to do anyone much good.

JDJ: Sun has been heavily criticized of late for apparently abusing the Java brand name and mislabeling non-Java software as Java. Do you think this is a risky strategy for Sun? Could it cause problems for the community?

AC: Absolutely. This is not what Java is about. It's so contrary to the spirit of what the Java community has been about, it almost feels unethical.

However, it doesn't matter that much. Sun has already ceded thought leadership on enterprise Java to BEA and others. Look at where the innovations are coming from. Take WebLogic Workshop, for example – that's real innovation. BEA and IBM share the overwhelming majority of the application server market. Customers who are deploying enterprise applications on Java today are doing it on either WebLogic or WebSphere, period.

It's ironic, actually, that Sun is taking the approach they are with the Java brand. For a company that's been so critical of Microsoft, what they've done is more reminiscent of Microsoft than it is of the company I've always thought Sun was. They have been a visionary leader in the past. I'm optimistic that they will be a visionary leader again in the future. But you don't get there with these kinds of tactics.

JDJ: You have invested significantly in building out the BEA suite of development tools. Why was this move necessary when the tools market was already quite saturated? What was missing that you needed to fill?
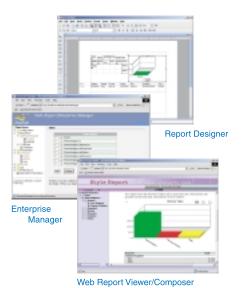
AC: We saw an opportunity to address a problem that other tools weren't addressing: unifying a development framework across every aspect of an enterprise application, from the business analyst to the J2EE expert. Java simply needs to be easier; if we are ever to achieve in the enterprise the economics and interoperability that the desktop has achieved, Java has to be easier. We delivered that prospect with WebLogic Workshop, in large part by breaking down a lot of the traditional barriers between different developers. Ultimately, customers will decide what role they want BEA to play and what role they want other software vendors to play. Our goal is to work in concert with the other tools vendors. We have made our platform as extensible as possible. Controls are just one example of how we've done this. ✐
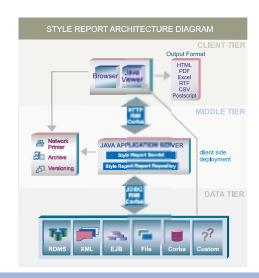
Java & Web Reporting:
with Java flexibility
without Java complexity

Style Report: Information Vantage Point

Style Report blends the best of Java and Web to provide an award winning Web enterprise reporting solution. Based 100% on open standards such as XML, JSP, Java and Web services, Style Report allows you fully leverage existing IT infrastructure and common technical skills. One Report, Multiple Channel delivery and zero client ad-hoc reporting gives end users the freedom and frees IT from time consuming repetitive work. Designed for integration, Style Report is deployed and managed as a standard Web application or Corba/RMI service requiring not special skills. Download a free evaluation copy now!

Report Designer

Enterprise Manager

Web Report Viewer/Composer

STYLE REPORT ARCHITECTURE DIAGRAM

CLIENT TIER

Output Format

Browser | Java Viewer

HTML
PDF
Excel
RTF
CSV
Postscript

HTTP
RMI
Corba

MIDDLE TIER

Network Printer
Archive
Versioning

JAVA APPLICATION SERVER
Style Report Servlet
Style Report Report Repository

client side deployment

JDBC
RMI
Corba

DATA TIER

RDMS | XML | EJB | File | Corba | Custom

Style Report Features

Feature	
Flexible Visual Designer with JavaScript or VB Script	✓
PDF, Excel, Word, CSV and other Export Format	✓
Database/XML and other Datasources	✓
Zero Client Browser Viewing	✓
Built-in Drilldown/Parameters	✓
Zero Client Ad-hoc Reports	✓
Virtual Private Security Model	✓
J2EE Drop-in Deployment	✓
Advanced Scheduling	✓
Report Versioning/Archive	✓
Demand Paging/Report Cache	✓
Portal/JSP integration	✓

InetSoft
open standards innovation

www.inetsoft.com
JDJ special http://www.inetsoft.com/jdj

Phone: 888-216-2353 (USA)
Phone: 732-424-0400 (Intl)

Which EII Solution
Is Right for You?

Managing integration tasks

Peter Chang

E
nterprise Information Integration (EII) represents a new category of software that enables disparate data silos to be integrated into a single virtual database for applications. This approach gives developers a powerful tool for simplifying data integration and building flexible applications. If you haven't heard of EII yet, you will soon as the industry rallies around this concept and more EII projects reach deployment.

This article explains why you should consider EII, describes its advantages for data integration, and discusses the different approaches to implementing EII. The article provides a framework for comparing EII products and choosing a suitable product to begin simplifying data integration in your environment.

Data Integration Issues

The presence of multiple, disparate data sources is a fact of life in enterprise IT environments. The number and types of data sources have only grown over time and made data integration a prominent aspect of IT investments.

To combat the inflexibility induced by information fragmentation, developers have many tools for data integration (see Figure 1). There are adapters for accessing data sources, transformation engines for reformatting data, and data warehouses for aggregating data from multiple sources. To integrate data sources, developers use these tools to program the integration requirements into applications.

Although this approach to data integration works, it requires a programmatic approach that has the following deficiencies:
- *Low-level programming:* The developer needs to create and maintain the integration code. The tools provide only the building blocks for integration. The developer must write the low-level code to implement the integration requirements, which greatly increases development and maintenance costs.
- *Multiple data source APIs:* Each data source has its own API and data format. The developer must understand each data source in sufficient detail to manage the data integration. This often requires multiple specialists to implement and maintain the integration, which again drives up complexity.
- *Inconsistent integration framework:* The developer has to manage all the integration issues such as the relationships between data sources and data formats. This leads to one-off solutions and inconsistencies that are difficult to maintain.
- *Tight-coupling:* The programmatic approach creates hard-coded dependencies between applications and data sources. This severely hampers maintenance since updating a data source can potentially break many applications. Even minor updates must be carefully considered and orchestrated.
- *Limited reusability:* The integration code tends to be application centric and data source centric. This limits the reusability of investments in data models and integration rules.

A Smarter Approach to Data Integration

A better solution for data integration is EII. EII supports data integration by enabling multiple data silos to be represented to applications as a single virtual database. Instead of integrating data in the application code, the data integration function is pushed out of the application layer into a new EII tier that sits between applications and data sources. This new tier is dedicated to managing integration tasks such as connecting to a data source, transforming data, and integrating data.

In this framework, the developer or analyst creates a logical data model in the EII server that represents the business view of information (aka the data integration requirements). The target data source's physical data models are mapped into the logical data model to create a virtual database schema. Applications interact with data sources through the EII server based on the logical data model. The EII server automatically translates application requests into queries against one or more data sources, integrates the data, and produces results according to the logical view.

EII's holistic approach to data integration addresses the drawback of the toolkit approach (see Figure 2). Instead of providing tools for each task, EII establishes a framework that automates the low-level details and exposes a high-level, declarative interface for specifying data integration requirements. Multiple data sources can be integrated without writing any application code. The developer defines the desired logical data model and maps the data source into the logical model using a GUI tool.

This end-to-end approach generates several advantages:
1. *No programming:* The declarative approach eliminates the need to create and manage integration code. The EII server automatically generates the low-level code to orchestrate the data integration.
2. *Unified API:* Applications access data sources through the EII server's API instead of through data source APIs. Developers do not need to know how every data source works.
3. *Automation:* The EII server manages the disparities between data formats and APIs. The developer uses the GUI tool to create the new data model and the EII server automatically generates the correct API calls and data transformations.
4. *Loose coupling:* Inserting an EII layer between applications and data sources effectively decouples the usual dependencies between them. The mapping between the logical data model and the data source simply needs to be updated to reflect any changes to the data source. This level of flexibility provides some interesting benefits:
   - An application can migrate to a different database of the same or different architecture.
   - A database schema can quickly evolve to reflect new business requirements.
   - Multiple versions of the logical data models can be created to support incremental system migration.

Peter Chang has been working with Java, middleware, XML, integration, and the Web in engineering and marketing capacities. Previously he was the director of product management at Raining Data and at Sonic Software. Peter is a graduate of the University of California, Berkeley.

pchang001@yahoo.com.

5. ***Common data model:*** A common data model can be established for the enterprise. It helps different groups understand and share the available information. The investment in a common model compounds over time to improve the strategic utilization of information assets.
6. ***Reusability:*** The logical data model can be aggressively reused in other projects. This reduces the need for additional data integration and centralizes ongoing administration. Reusability changes the data integration focus from code management to strategic information management.

Approaches to EII

How do you get started with EII? Like all emerging markets, EII offers many different approaches to implementing the solution. In general, the available products can be differentiated based on the underlying logical data model, the data transformation framework, and the query interface.

Logic Data Model

The logical data model is at the heart of every EII solution. Physical data models from target data sources are mapped into the logical data model. This model serves as the schema of the virtual database. The types of logical models include relational, object, and XML. Depending on the approach, the EII servers would appear to applications as an object database, a relational database, or an XML database.

Transformation Framework

The transformation framework dictates how data from multiple sources are transformed into the logical data model. The approaches to transformation include SQL, XQuery, and proprietary. A visual data mapping tool is typically provided to shield the developers from low-level transformation code; however, if the mapping tool doesn't accommodate specific integration requirements, the developer may need to edit or write transformation code directly.

Query Interface

The query interface provides access to data. It dictates how applications read, query, insert, update, and delete the underlying data represented by the logical data model. It also dictates the data format presented to applications. EII query interfaces include SQL, XQuery, and proprietary.

Although the data model, transformation framework, and query interface can be mixed and matched, in practice, specific transformation frameworks and specific query interfaces work best with specific data models. Therefore, the general approaches to EII can be grouped into three main categories: relational, object, and XML (see Table 1).

Relational Approach

Products that take the relational approach use the relational data model as their logical data model. All data sources are represented as a series of tables, and SQL is used to transform the data into the logical model. With this approach, the EII server is a virtual relational database and applications use SQL to interact with the integrated data.

MetaMatrix is an example of this approach.

Object Approach

Products that take the object approach use objects as the logical data model. All data sources are represented as objects, and automatically generated code is used to transform the data into the logical model. With this approach, the EII server is a virtual object database and applications use a proprietary interface to obtain data objects.

Journée is an example of this approach.

XML Approach

Products that take the XML approach use XML as the logical data model. All data sources are represented as XML document collections and XQuery is used to transform the data into the logical model. With this approach, the EII server is a virtual XML database and applications use XQuery to interact with the integrated data.

BEA Liquid Data for WebLogic, Ipedo, and Nimble are examples of this approach.

Each approach to EII has its advantages and disadvantages. The primary distinctions are based on the data modeling flexibility, the query flexibility, and the result-processing requirements. These differences affect the suitability of each approach to specific applications and developer predilections.

Data Modeling Flexibility
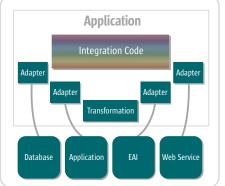
The distinction in modeling flexibility deals with the expressiveness of the data model that can be created from multiple data sources (see Table 2). This is largely dependent on the underlying logical data model and the ability of the transformation framework to mold data structures into the new logical model.

In this regard, the object and XML approach have an advantage over the relational approach because of their ability to support hierarchical data relationships. Their logical data model can directly represent hierarchical data whereas the relational approach must decompose the data structure into tables. In addition, the object and XML approach can represent hierarchical data sources like XML, whereas the relational approach requires the developer to reconstruct the hierarchical data in application code.

This modeling advantage is important to applications that use data from nonrelational data-sources sources such as message queues, Web services, XML documents, EJBs, and applications.

Query Flexibility

Another key distinction between each approach is query flexibility. This deals with the level of filtering and data processing that can be specified in a query and executed by the EII tier.
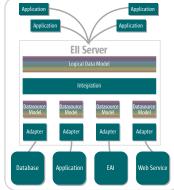
Figure 1 Traditional data integration

Figure 2 EII for data integration

Approach	Data Model	Transformation	Query
Relational approach	Relational	SQL	SQL
Object approach	Object	Proprietary	Proprietary
XML approach	XML	XQuery	XQuery

Table 1 Summary of characteristics of different approaches to EII

The SQL and XQuery query languages have an advantage over simply retrieving objects by criteria. The language approach better supports projection, aggregation, and joins, which enable fine-grained results to be produced by a query. In contrast, query returns of object collections may generate unnecessary data and require additional processing in application code to aggregate and join data.

XQuery has a further advantage over SQL and Object with its enhanced data manipulation facilities. It provides a functional programming language that can express complex transformations against any data structure. It supports built-in and external functions, conditional processing, scripting, and the ability to transform results into any text or binary format.

The distinction in query flexibility is important because the more data processing the EII server can perform, the less code the developer needs to create and maintain. Query flexibility also has implications on performance. A query that returns just the desired data, minimizes network traffic and improves the response time. To achieve the smallest result set, the query must maximize the level of data selection, projection, aggregation, joins, and transformation that can be expressed and processed in a single query.

Result Processing

The last primary distinction between the three approaches is the result-processing model. The relational approach generates tabular data, the object approach generates native program objects, and the XML approach generates XML documents.

The object approach has an advantage over the relational and XML approach since a native object representation of data is the most convenient to work with. The developer directly accesses the data by simply calling the specific data object's methods. In the relational and XML approach, the developer must work with generic data structures such as JDBC ResultSets or DOM objects. These require additional work over the object approach to read, update, create, and delete data.

Selecting an Approach

The capabilities and limitations of each approach affect their suitability to particular environments, data processing requirements, and data source integration requirements. The best approach truly depends on the unique set of requirements. The following are some general guidelines for selecting an approach.

The relational approach works very well with established data programming practices. Developers can get started quickly and leverage traditional techniques and know-how. Although the relational data model is less flexible, the majority of enterprise data resides in relational databases and many nonrelational data sources can be coaxed into a relational format. The pain of accommodating a few nonrelational data sources may be outweighed by the approachability of relational development.

The object approach provides a flexible, logical data model for integrating diverse data sources. For complex environments, this capability greatly simplifies data integration and produces detailed views of enterprise data assets. In addition, the object interface makes data programming more conven-

ient. Fans of object-to-relational mapping tools and object databases will appreciate the object approach. Unfortunately, binding data to program objects is also the weakness of the object approach. This makes query processing less flexible and requires the application developer to process more data. The resulting inefficiency makes the object approach unsuitable for ad hoc query processing where the data binding can't be tuned for performance.

The XML approach is the most cutting-edge architecture for data integration. Using XML as the logical data model and XQuery as the query interface provides a flexible platform for EII. XML effectively models many enterprise data sources and XQuery provides powerful data processing capabilities. Together these minimize the integration code in applications. These advantages are the most apparent in an environment with heterogeneous data sources and mixed application architectures. Although the XML approach provides a great deal of flexibility, the technology is less mature and requires a shift toward XML-centric application development. For developers ready to experiment and blaze new trails, the payoff will be worthwhile.

Other Considerations

The comparisons between different approaches to EII provide a good starting point for exploring the market offering, but they don't tell the whole story. In addition to exhibiting many different approaches to the solution, emerging markets also exhibit products with different levels of feature coverage. This affects the capability and performance of EII products beyond the architectural differences. Understanding the following key features will help to further narrow the list of qualified candidates.

Adapters

Also known as connectors, these software components make a data source available to the EII server. This is the first item of any checklist since there is no solution if your target data sources are not supported. Most vendors get the bulk of their adapters from third-party providers. The level of integration between the EII server and the adapter can vary, so look for adapters that are well integrated into the graphical data mapping tool and the query interface.

If your target data source is not supported, most vendors offer a development kit for building custom adapters. The effort will vary but it won't be trivial, so consider this option carefully.

Update

In addition to accessing data, the ability to propagate updates to the underlying data sources is critical. A bidirectional EII solution completes the virtual database illusion to keep the application decoupled from data integration. Look for the ability to update logical data entities made from multiple data sources and look for strong transaction management capabilities.

Security

The logical data model in an EII system will cross many different data sources, so a good security model is needed to manage access to the integrated information. Look for fine-grained access management and integration with enterprise directory servers like LDAP directories.

Caching

Integrating data from multiple sources can be computationally expensive. An EII server's query response time can be

Distinctions	Relational	XML	Object
Data modeling capability	OK	Excellent	Excellent
Query capability	Good	Excellent	OK
Result processing	ResultSet	DOM/SAX	Object

Table 2 Summary comparison of distinctions in different approaches

an issue depending on the number of data sources, the number of joins, and the level of transformation required to generate a result according to the logical data model. Caching query results is a good strategy for reducing the processing requirements and improving performance.

There are two kinds of caches. The more conventional is a read cache that saves prior query results. The EII server scans the cache for past results identical to the query before going to back-end data sources and building new results. To avoid stale data, a read cache periodically clears the cached results. Alternatively, the read cache can also be synchronized with the source, which enables cached results to be selectively cleared or refreshed. This keeps the cache hit rate high to reduce query response times.

The other type of cache is a query cache. This approach caches all the data in the logical data format and directly processes arbitrary queries against the cached data. A query cache is like a data mart except with active synchronization. This approach eliminates all the physical-to-logical data mapping and all queries against the underlying data sources during a request. For sophisticated logical data models, a query cache can greatly improve performance. However, a query cache's storage and synchronization requirements can be enormous. Careful data partitioning is critical to keep the cache size manageable and filled with sufficiently current data.

A read cache and a query cache can be used in combination to better match data usage requirements. A read cache is best for situations with predictable and repetitive queries, while a query cache is best for environments with ad hoc queries against relatively static data sources.

Summary

Enterprise information integration is still very new but the value proposition is realizable. It can dramatically reduce the complexity of working with multiple data sources and produce much more flexible applications.

A starting point for evaluating the suitability of EII products is to consider the three general approaches to EII: relational, object, and XML. Each approach has its advantages and disadvantages, so the best solution depends on the specific requirements. In general:
- The XML and object approaches provide greater data integration capabilities but do not fit as well into traditional database application development environments.
- The XML and relational approaches provide more flexible query facilities but require more work than the object approach to process query results.
- XML with XQuery provides the most powerful modeling and querying facility but the technology is still evolving.

Once the choices have been narrowed, consider the availability of adapters, support for updates, the security model, and the caching model to make the next cuts. This should identify a suitable product to begin leveraging EII. ⬮

References
- *BEA Liquid Data for WebLogic:* www.bea.com
- *Ipedo:* www.ipedo.com
- *Journée:* www.journee.com
- *MetaMatrix:* www.metamatrix.com
- *Nimble:* www.nimble.com
- *XQuery:* www.w3c.org/XML/Query
- *Raining Data:* www.rainingdata.com

Where Is *i-*
Going in

Scott McNealy on network computing, virus attacks, Linux,

McNEALY EXCLUSIVE:

"Network computing is at a tipping point, as the race to connect everything of value is driving widespread adoption of innovations like Java technology, and hundreds of millions, and soon billions, of devices get on the network and need to share information securely and reliably."

by **Scott McNealy**

With new technologies like radio frequency identification (RFID), many tedious, time-consuming tasks will disappear from our daily routine. Instead of waiting in line 30 minutes to check out at the supermarket, we'll just push the cart past a scanner and out the door, and we'll get a bill for the groceries at the end of the month.

Technology like N1 – designed to automate management, virtualization, and provisioning in the data center – will break down the traditional barriers of cost and complexity, driving far greater return on IT investments than ever before. The adoption of grid computing will grow by leaps and bounds as IT professionals seek ways to get serious computing power from low cost components.

For over a decade we've said "the network is the computer," and that has never been more true than it is today. The world has embraced a vision of network computing based on industry standards and open interfaces. Every country on the planet, every industry, and hundreds of millions of consumers are enjoying the benefits of the Java lifestyle. It's all about write once, run anywhere, and it couldn't be any simpler as mobile phones, PDAs, gaming consoles, cars, and even that bag of potato chips in the supermarket become part of a network that will eventually connect trillions of things.

-Technology

n 2004?

and software licensing in 2004

2004 is going to be a very dynamic year, with all kinds of innovation happening in network computing. And I'm sure I'll see you all on the network…

2004 Predictions

1. Everyone and everything with a digital, electrical, or biological heartbeat, and even inert objects, will continue to be connected to the network in growing numbers. Nearly half a billion Java Cards have shipped so far. Entire countries such as Belgium and Taiwan are issuing them to every citizen to enable universal, secure, single sign-on access to the network. Java Card technology is revolutionizing authentication and access to health care services, banking, and other applications and services on the network.

2. 2004 just may be the year when a virus or some other kind of attack causes a network outage on a global scale that could do some real damage to the economy. Sobig-F, Blaster, and other worms and viruses did huge damage out there in 2003, as ATMs, airlines, and railroads were forced to "reboot." The days are over when a general purpose operating system is secure and reliable enough for these kinds of applications. The smart folks are going to take 2003 as a wake-up call, and in 2004 we'll see even more enthusiasm for Java technology and its widely recognized security model.

3. The world is still going to keep a tight hold on its purse strings, and every dollar will be hard earned. There's no magic pill that's going to bring about an economic recovery. For the tech sector, it's going to take a complete rethinking of IT investments in terms of competitive advantage, operating and maintenance costs, and total cost of ownership.

JEREMY ALLAIRE

JOHN BOSAK

TIM BRAY

MARC FLEURY

DAVID LITWACK

RAJIV GUPTA

VIJAY TELLA

BILL RAY

RICK ROSS

MITCHELL KERTZMAN

ADAM KOLAWA

SAMUEL J. GREENBLATT

Jason Bell
Core and Internals Editor

Exposing **the Services**

Depending on who you talk to, the response you get when you mention the words "Web" and "services" in the same sentence can vary from a big smile to an amazingly serious frown. It's easy to develop an application or Web site that uses the Amazon API and the Google API to great effect. From where I stand there hasn't been much movement since, although personal Web publishing has taken to the Blogger API with some interesting results. Syndic8 also has a nice API to expose the collection of news feeds it has, as does Meerkat. Once again the community is reduced to a select few using a good quality tool. I'd be interested to find out the usage of these APIs, so if anyone could help me out, please e-mail me.

Don't get me wrong, I know there are lots of Web services out there. Look at XMethods' Web site to get a very long list of active services. The question is, how many of those would you use on a day-to-day basis? I have no need to search and find Icelandic addresses or social security numbers.

What now for Web services? Amazon looks like it is taking the whole thing seriously enough. Jeff Barr, the Web services evangelist, defines four sets of users: buyers, sellers, site owners, and developers. They are working hard to make their API as robust as possible so developers can craft some quality software. With Google, however, I can't quite put my finger on it. They released the API (a month after I had spent a lot of time scraping URLs for an IRC bot plugin) that worked very well for doing searches, spelling suggestions, and taking data from the cache. There's still no word on an API for the comparison shopping engine, translation services, or the excellent news services. Has the Google API stagnated? I do hope not.

What can the Java community do to help? In fact we can do quite a lot. Our point of focus though has to be the end user, not the developer. Well-thought-out designs, processes, and responses need to be addressed for Web services to work in the real world. Programming Web services is a simple process; programming robust ones isn't. Just because users can't see what's going on with our Web service code,

> " Web services are still young but they have a lot of potential to expose your services to other clients and users. **Development still has to come from the community, and I think we can do it**"

One thing a lot of the mainstream companies seem to suffer from is a lack of completion from search to sale. If I want to get details on a flight from London to New York, I can do so quite easily via a Web service. Booking a ticket still takes a phone call or me logging on to a Web site to finalize the sale. The whole thing could be wrapped up in three or four steps (including payment). There should also be an amount of intelligence wrapped in – if I can't fly from London Heathrow to New York, the Web service should be suggesting alternative airports that can. Most of this functionality is down to the server side. What I have noticed is a standard blank response to the most basic of requests. The majority of humans don't make requests in that way; software shouldn't either.

I've been thinking about Web services a lot recently. I've been talking to a company that provides an excellent service and we've worked closely over the last six months on resolving some request for information that would make my life easier. To their credit they came up with the goods very quickly. Then I started talking about how Web services could improve their software model. At present everything, with a few exceptions, is done via their Web site. The software I had written bypassed the Web site but still used HTTP. The implementation was okay but not what I would like to put my name against for the rest of my days.

doesn't give us license to be sloppy developers. Aim for the highest quality every time. Test it, unit test it, and stress test as many times as you can. Don't just write the service, write the client code too, in a number of languages. Be a help to these people, and realize that offering a bunch of WSDL files may not be the way to attract users.

Web services are still young, but they have a lot of potential to expose your services to other clients and users. Development still has to come from the community, and I think we can do it.

JDJ News

What do you think about the new layout? My plan, as you may have noticed from the topics in this month's section, is to concentrate on a fairly pragmatic approach – the information you need presented to you in the easiest way possible. This is not easy to ask or deliver but I am working on it. Over the next few months you'll see articles on refactoring, more beginners tutorials, and also one-page articles covering just what you need to know to get the job done.

The Web site will offer more content as well, plus lively discussions about various topics. Don't forget, if you have a RSS reader, you can link to our feeds, and it covers various sections. This will give you the news on your desktop (still visit the site once in a while though, please).

Jason Bell is a technical architect for a business intelligence company in England. He is also involved in a number of open source projects and reads the API docs.

jasonbell@sys-con.com

WILL IT WORK?
SHOULD YOU EVEN HAVE TO ASK?

Data connectivity is vitally important to the health of your applications. Is this really where you want to take chances? Your data connectivity choices can have dramatic effects on scalability, interoperability and performance. And you'll be left facing increasing development, maintenance and testing costs plus potential loss of revenue. DataDirect offers the industry's most comprehensive, proven suite of database-independent Type 4 JDBC drivers. Our extensively tested J2EE-certified drivers include the most advanced JDBC 3.0 features including Distributed Transaction Support, Connection Pooling and BLOB/CLOB support.
DataDirect Connect® *for* JDBC® is the SPECjAppServer and ECperf performance leader.

Find out what else you might be missing. Download our whitepaper,
"What you don't know about database drivers CAN hurt you" @ www.datadirect.com/JDJ

DataDirect™
TECHNOLOGIES

www.datadirect.com
800-876-3101

Javassist: **Java Bytecode Engineering Made Simple**

Shigeru Chiba

For source-level abstraction

Javassist is a powerful new library in the field of bytecode engineering. It allows developers to add a new method to a compiled class, modify a method body, and so forth. Unlike other similar libraries, Javassist enables this without knowledge of Java bytecode or the structure of a class file.

Bytecode engineering is used to manipulate and modify compiled Java classes and to programmatically create new classes. This engineering can happen either at runtime or at compile time. Some technologies use bytecode manipulation to optimize or enhance existing Java classes. Other technologies use it to make themselves easier to use or to avoid cumbersome code generators. For instance, the JDO 1.0 (Java Data Objects) specification requires bytecode enhancement of pre-compiled Java to add database persistence code to simple Java classes. In Aspect-Oriented Programming, some of the new frameworks use bytecode engineering to enhance Java classes with cross-cutting functionality. EJB containers, like JBoss, dramatically speed up the development cycle by dynamically creating Java classes at runtime to avoid the costly code generation/precompilation step required by EJBs. Even the JDK does bytecode manipulation within its java.lang.reflect.Proxy class.

Bytecode manipulation has been a complex and often unacceptable task for framework developers because of the huge implementation costs. Learning bytecode is very similar to learning an assembly language. The learning curve can be very steep for developers. Not only that but code written to engineer bytecode is a challenge to maintain, as it's difficult to read and follow. Javassist is a library for simplifying bytecode manipulation. It allows developers to fully exploit bytecode manipulation with little or no knowledge of bytecode, giving them a degree of fine-grained control.

API Parallel to the Reflection API

The first part of the Javassist API is similar to the Java Reflection API. It allows you to view the structure of a Java class before it is loaded by the ClassLoader. It includes CtClass, CtMethod, and CtField classes as the Reflection API does java.lang.Class, java.lang.reflect.Method, and Field. These classes represent a class, a method, and a field that have not been loaded yet. They provide a number of methods parallel to the Reflection API, for example, getName, getSuperclass, getMethods, getSignature, and so on. The following code reads org.geometry.Point.class, analyzes the definition, and prints the name of the super class (in this article, we always assume that the javassist.* packages are imported):

Shigeru Chiba is an associate professor at Tokyo Institute of Technology and a member of the JBoss development team.

chiba@acm.org

```
1. ClassPool pool = ClassPool.getDefault();
2. CtClass pt = pool.get("org.geometry.Point");
3. System.out.println(pt.getSuperclass().getName());
```

The ClassPool object is a factory of CtClass objects. It searches for a class file in the specified class path and creates a CtClass object as a singleton for each class. The get method in ClassPool returns the CtClass object representing the class with the given name.

The Javassist API does not provide methods for creating a new instance (the newInstance method), invoking a method (the invoke method), or accessing a field value (the get and set methods) since CtClass objects represent classes that have not been loaded. On the other hand, the API provides methods for changing class definitions. For example, setSuperclass in CtClass changes the super class of the class. The reflection API does not support such changes. For example:

```
4. pt.setSuperclass(pool.get("Figure"));
```

This modifies the definition of the Point class so that it extends the Figure class. For consistency, this example assumes that the Figure class is compatible with the original super class.

Adding a new method to the Point class is also possible:

```
5. CtMethod m = CtNewMethod.make("public int xmove(int dx) { x +=
   dx; }", pt);
6. pt.addMethod(m);
```

The CtMethod object that represents the added method is created from the given source text. Developers do not have to write a sequence of virtual machine instructions by hand. Instead, Javassist compiles the given source text in Java with a custom compiler included in Javassist.

Finally, to reflect all the changes above, the writeFile method is called:

```
7. pt.writeFile();
```

The writeFile method in CtClass writes the modified definition of the class to a class file. Javassist can work with a ClassLoader, as we'll discuss later.

Javassist is not the first class library for writing a bytecode translator. For example, Jakarta BCEL is a popular library for bytecode engineering. However, you can't use Jakarta BCEL with source-level vocabulary. If you add a new method to a compiled class, you must specify a method body by a sequence of bytecode instructions. On the other hand, Javassist allows you to specify it by source text as shown above.

Instrumenting a Method Body

The instrumentation of method bodies can also be described with source-level vocabulary. Developers do not have to directly manipulate virtual machine instructions. Although the range is limited, Javassist enables various typical instrumentations. If developers need fine-grained instrumentation that they cannot describe with source-level vocabulary, they can use the low-

level part of the Javassist API, which is not described in this article. It's a similar bytecode-level API to Jakarta BCEL.

The design of the Javassist API for instrumenting a method body is based on the idea of Aspect-Oriented Programming (AOP). Javassist allows the identification of some expressions, such as method calls and field accesses, in a method body and then substitutes a code fragment for the expression.

For example, Listing 1 first obtains a CtMethod object that represents a draw method in the Screen class; then it searches the body of the draw method for method calls to move in Point class. All the occurrences of the method calls to move are replaced with the following block statement:

```
{ System.out.println("move"); $_ = $proceed($$); }
```

so that a message is printed out before a move is called. The statement, written in special syntax:

```
$_ = $proceed($$);
```

executes the original method call with the original parameters.

The instrument method in CtMethod searches the method body and, when it finds a method-call expression, calls the edit method on the given ExprEditor object. The parameter to the edit method is a Method-Call object representing the found expression. Since this object provides methods for getting static properties of the expression, the edit method first examines the name of the called method and, if the method is moved in Point, it replaces the method-call expression with the block statement printing a message. The block statement is compiled into bytecode by Javassist before the replacement.

Special Variables

In the substituted statement, several special variables beginning with $ are available. For example, $_ represents the result value of the substituted expression. $$ represents a list of the original parameters to the method called in the replaced expression. $proceed represents the name of the originally called method.

Each parameter to the originally called method is represented by $1, $2, $3,…. The target object of the method call is represented by $0. These special variables are useful when developers want to change some of the parameters. For example, if the substituted block given to the replace method at line 8 was as follows:

```
{ System.out.println("move"); $_ = $proceed($1, 0); }
```

then the second parameter to the move method would be zero.

Another special variable $args is an Object array containing all the parameters to the originally called method. If the type of a parameter is a primitive type, then the wrapper object containing that parameter value is stored in the array. For example, if the parameter is an int value, the java.lang.Integer object containing that int value is stored in the array. $args is useful when the parameters are used to call a method through the invoke method in java.lang. reflect.Method.

Javassist also allows the insertion of a code fragment at the beginning or end of a method body. For example, the insertBefore method inserts the given block statement at the beginning of the method body.

```
1. ClassPool pool = ClassPool.getDefault();
2. CtClass cc = pool.get("Screen");
3. CtMethod cm = cc.getDeclaredMethod("draw", new CtClass[0]);
4. cm.insertBefore("{ System.out.println($1); System.out.print
   ln($2); }");
5. cc.writeFile();
```

This example inserts a code fragment at the beginning of the draw method so that the values of the two parameters to draw are printed out. $1 and $2 are special variables representing the first and second parameters to draw.

Special variables starting with $ are also available in the source text passed to the setBody method in CtMethod. The following example shows how to add a wrapper method:

```
1. CtClass cc = sloader.get("Point");
2. CtMethod m1 = cc.getDeclaredMethod("move");
3. CtMethod m2 = CtNewMethod.copy(m1, cc, null);
4. m1.setName(m1.getName() + "_orig");
5. m2.setBody("{ System.out.println("call"); return $proceed($$);
   }",
            "this", m1.getName());
6. cc.addMethod(m2);
7. cc.writeFile();
```

This program first makes a copy of the move method in the Point class, then it renames the original move method to move_ orig. It then changes the body of the copy of the move method so the copy will be a wrapper method named move, which prints a message and invokes the move_orig method. The second parameter to setBody specifies the target object of the $proceed call, and the third parameter specifies the name of the method called by $proceed.

Javassist also has a multitude of other helper objects and functions that allow you to do things like replace field access, redirect method calls, and simplify how you insert code before or after a method. Check out www.javassist.org for more information and tutorials.

Performance Overhead

Some developers seriously care about the runtime performance of class files modified by Javassist. However, runtime penalties due to Javassist are extremely low, thanks to the custom compiler included in Javassist. While handling special variables starting with $ is the major source of the runtime penalties, the custom compiler produces optimized bytecode to reduce the overhead. For example, developers can use the instrument method in CtMethod to replace a method-call expression. If the substituted block statement is equivalent to the original expression, only a few extra machine cycles will be paid at runtime for every method call at that expression.

ClassLoader

Javassist can be used in conjunction with a ClassLoader to modify the bytecode of a class at runtime before it's loaded. In Java, developers can customize the classloading behavior by implementing a custom ClassLoader. They can use this mechanism so that class definitions will be modified on demand with Javassist when the classes are loaded.

The simplest way to modify and load a class at runtime is to call the toClass method in CtClass instead of the writeFile method. In the previous examples, the modified class files have been written out on a disk. However, the toClass method loads the class by a custom ClassLoader of Javassist and it returns the java.lang.Class object that represents the loaded class. Listing 2 dynamically defines the Hello class from scratch and adds the say() method, then it loads the Hello class and calls that method.

The definition of the IHello interface is as follows:

```
1. public interface IHello {
2.   void say();
3. }
```

Tips on ClassLoaders

Using toClass is simple but developers must be careful. Because of Java's classloading algorithm, developers might be confused and run into ClassCastExceptions. For example, in Listing 2, let's load the IHello interface with the toClass method before loading the Hello class, resulting in:

```
       :
 7. Class ih = ci.toClass();
 8. Class h = cc.toClass();
 9. IHello obj = (IHello)h.newInstance();
10. obj.say();
```

This change makes the cast operation at line 9 throw a ClassCastException, since the IHello interface appearing in the definition of the Hello class is different from the IHello interface at line 9.

To understand this situation, you must know that multiple ClassLoaders can coexist in Java and they form a tree structure. Each ClassLoader except the root has a parent ClassLoader, which normally loads the class of that child ClassLoader. The request to load a class is delegated along this hierarchy of ClassLoaders.

If a class file is loaded by two distinct ClassLoaders, it becomes two distinct classes with the same name and defi-

nition. Since the two classes are not identical, an instance of one class is not assignable to a variable of the other class. In the previous example, IHello is loaded at line 7 by a custom ClassLoader of Javassist (let's call it LJ). Since the loaded classes and interfaces are cached in the Class-Loader, the IHello interface loaded by LJ is used as the interface implemented by the Hello class when it's loaded by LJ at line 8.

On the other hand, IHello at line 9 represents the interface that's loaded by a ClassLoader that loads the class including line 9 (let's call it LP). Since this ClassLoader LP also loads the class of LJ, it's the parent ClassLoader of LJ. Since the object created by h.newInstance() has the IHello interface loaded by LJ but not LP, the cast operation at line 9 fails and throws an exception. Note that if line 7 were removed, LJ would delegate at line 8 to the parent Class-Loader LP to load IHello, and thus the Hello class loaded by LJ would implement IHello loaded by LP. Thus the cast operation at line 9 would succeed.

To avoid this ClassCastException problem, Javassist provides another ClassLoader that allows developers to fully control the classloading behavior. Developers can define an event listener that is notified every time a client requests to load a class so that the listener can properly modify the class.

Summary

Javassist is a powerful Java library that helps provide instrumentation of bytecode at load time or at compile time. Since it provides source-level abstraction, which is higher than the abstraction by similar libraries, using Javassist is relatively easy and detailed knowledge of Java bytecode is unnecessary. ✐

Resources

- *Javassist:* www.javassist.org
- *JBoss:* www.jboss.org
- *Jakarta BCEL:* http://jakarta.apache.org/bcel
- Chaudhri, R. "Understanding the Java Classloading Mechanism." *Java Developer's Journal*, Vol 8, issue 8.
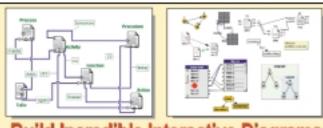
Listing 1
```
1. ClassPool pool = ClassPool.getDefault();
2. CtClass cc = pool.get("Screen");
3. CtMethod cm = cc.getDeclaredMethod("draw", new
   CtClass[0]);
4. cm.instrument(new ExprEditor() {
5.   public void edit(MethodCall m)
     throws CannotCompileException {
6.     if (m.getClassName().equals("Point")
7.       && m.getMethodName().equals("move"))
8.       m.replace("{ System.out.println(\"move\"); $_ =
         $proceed($$); }");
9.   }
10. });
11. cc.writeFile();
```

Listing 2
```
1. ClassPool pool = ClassPool.getDefault();
2. CtClass ch = pool.makeClass("Hello");
3. CtClass ci = pool.get("IHello");
4. ch.addInterface(ci);
5. CtMethod m = CtNewMethod.make(
     "public void say() { System.out.println(\"Hello\");
       }", ch);
6. ch.addMethod(m);
7. Class h = ch.toClass();
8. IHello obj = (IHello)h.newInstance();
9. obj.say();
```

WebAppCabaret ™

J2EE Web Hosting

www.webappcabaret.com

Quality Web Hosting at a reasonable price...

<JAVA WEB HOSTING AND OUTSOURCING>

You have developed the coolest mission-critical application. Now you need to deploy it.
Outsource your hosting and infrastructure requirements with WebAppCabaret so you can save time and money and concentrate on other important things. WebAppCabaret is the leading JAVA J2EE Web Hosting Service Provider. From shared hosting to complex multi-dedicated server hosting, WebAppCabaret has the right solution for you. 30 Day Money Back Guarantee and SLA.WebAppCabaret offers the latest Standards based Servlet containers, EJB servers, and JVMs. We provide options such as e-Commerce, EJB 2.x, Failover, and Clustering. Our Tier 1 Data Center ensures the best in availability and performance.

At WebAppCabaret you have the flexibility to choose the LATEST hosting technology best suited for your
WEB application or your programming skill. If you are a programmer or consultant,WebAppCabaret has
the right hosting solution for your project or client's dynamic web application/services requirements.

OUTSOURCING:
Do you really need an IT department for your web applications, mail systems, and data backups when we can perform the same functions more efficiently at a fraction of the cost - with competent technical expertise and redundant hosting facilities.

J2EE HOSTING:
Below is a partial price list of our standard hosting plans. (Reseller accounts also available). For more details please log on to http://www.webappcabaret.com/jdj.jsp

$39/mo Enterprise

Latest JBoss/Tomcat/Jetty
Latest JSP/Servlets/EJBs
Private JVM
Choice of latest JDKs
Dedicated IP Address
NGASI Control Panel
PHP and Perl
Web Stats
900MB Disk
200MB DB
MySql
PostgreSql
Dedicated Apache
Telnet . SSH . FTP
5 Domains
100 Emails
Web Mail . POP . IMAP
more...

$191/mo Dedicated

Managed Dedicated
Server starts at $191
per month for:
256MB RAM
Pentium 4
40GB RAID
Firewall
Unlimited Domains
Unlimited Web Site Hosting
NGASI Control Panel
with your own Logo
Each dedicated
server configured
for standalone
web application
and web hosting (resellers)
at no extra charge.
more...

$17/mo Professional

Latest Tomcat/Jetty
Latest JSP/Servlets
Private JVM
Choice of latest JDKs
NGASI Control Panel
PHP and Perl
Web Stats
200MB Disk
30MB DB
MySql
Telnet . SSH . FTP
2 Domains
20 Emails
Web Mail . POP . IMAP
more...

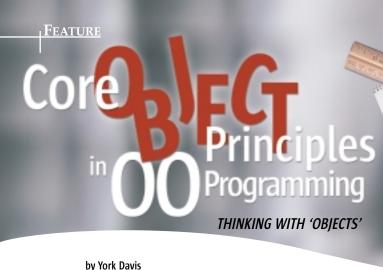### $99/mo Reseller

If you cannot afford a
Dedicated server for
reselling web hosting,
for $99/mo the Shared
Reseller plan gives you:

5 Professional Plans
10 Basic Plans
NGASI Control Panel
with your own Logo

more...

Core OBJECT Principles in OO Programming

THINKING WITH 'OBJECTS'

by York Davis

ndividuals just starting out with Java and object-oriented (OO) programming often feel overwhelmed by not only having to learn a new language syntax but also having to comprehend the unfamiliar concepts of OO programming. For those individuals, building a strong foundation on the key OO concepts is a great place to start. For seasoned developers, a refresher on these topics is always beneficial.

Objects

The most basic concept in an OO language is that of an object. In OO languages, objects represent real everyday items such as an automobile, radio, or dog. Objects can contain both attributes and behaviors. For instance, an automobile has the concept of a manufacturer name attribute and the concept of a drive behavior. In Java, attributes are represented as variables while behaviors are represented as methods. Variables can be a variety of different types – simple types such as int or complex types such as another object. Listing 1 shows an Automobile class.

For a real Automobile object, we would certainly not limit ourselves to such a small list of attributes and behaviors. In fact, our Automobile object should contain all of the properties and methods that a real-world automobile would actually have. We can therefore define an object as all of the attributes and behaviors that a real-world entity would possess. Objects can be used to represent more abstract concepts or entities as well. For instance, you could create an object to represent an idea.

One common area of confusion is the differences and similarities between classes and objects. In all OO languages, classes become objects when they are instantiated. In Java, object instantiation is performed using the new operator, as in the following line of code.

```
Automobile auto = new Automobile();
```

Until a class becomes an object through instantiation, it's only a shell of what the object can be. Think of a class as an object blueprint. Just as a house exists only in blueprint form until it is actually constructed, classes do not become objects until they are constructed via instantiation. Once an object is instantiated, it contains its own version of all the instance methods and instance variables of the class. I'll be using the terms class and object interchangeably.

York Davis is a senior managing consultant at Software Architects, Inc. With more than 12 years in the software development field, York has been using Java for more than five years and has extensive experience building enterprise application architectures.

ydavis@sark.com

Objects communicate by calling each other's methods. This concept is called messaging. To send a message is simply to invoke a method of a called object by a calling object. Depending on the method signature, messages can include one or more parameters that can affect the operation of the method. Parameters can be simple types like an int or complex types like another object. As an example, perhaps our Automobile object contains an accelerate() method that accepts an int value representing the speed to which the Automobile object should accelerate. A Driver object, representing the operator of the automobile, can pass a message to the Automobile object's accelerate() method causing the vehicle to increase its speed. A message can optionally return a value. As with the parameters sent to a message, a message's return value type can be simple or complex.

The following example illustrates the concept:

```
public class Driver {
 public static void main(String args[])
 {
  Driver driver = new Driver();
  driver.drive();
 }

 private void drive() {
{
 Automobile auto = new Automobile();
 auto.accelerate(25);
}
}
```

I'll be using the terms message and method interchangeably.

Coherence

Coherence is the concept of modeling objects in such a way that they reflect, as accurately as possible, the actual properties and behaviors of real-world entities.

This goal, though desirable, is not always easy to achieve. Even experienced object modelers struggle with the balance between making an object do too much or too little and assigning responsibility to the correct object. Sometimes, however, a class ends up with functionality that is not central to the core concept of the class. Let's return to our Automobile class for another example. Let's pretend that a request was made for the capability to calculate the price of fuel in different international currencies. Would it make sense to add this method to the Automobile class? Certainly it

wouldn't break the Automobile class by doing so. However, the object would now be somewhat fragmented in terms of its functionality. You could say that the Automobile class would be less coherent.

There's no doubt that performing this calculation is related to the processes surrounding using an automobile. But doing so is not central to what an Automobile object does. Generally, if a property or behavior is not essential to the concept that the object is attempting to model, it does not belong in the object and will reduce the object's level of coherence. Reducing the level of coherence in an object also reduces the intuitive understanding of what a real-world entity does and how it behaves. A better alternative is to place the fuel price conversion method in a different class – perhaps a Trip or TripUtilities class.

Inheritance

Inheritance provides the capabilities for one object to automatically contain attributes and behaviors of another object. Let's quickly return to our Automobile object and show an example of inheritance, how it works, and why it's so important. Perhaps in addition to our Automobile object, we wanted to create a new object DriversEducationAutomobile. DriversEducationAutomobile is exactly like Automobile in every way except that it has an extra brake pedal on the front passenger side for the instructor to depress.

One option in creating DriversEducationAutomobile is to copy all of the attributes and methods of Automobile, add the new attributes and methods that are specific to DriversEducationAutomobile, and create a new source and class file. That will certainly work but would create a bit of a maintenance problem. Now if we wanted to make a change to any of the common variables or methods shared between Automobile and DriversEducationAutomobile, we would need to make changes in both classes. A better way is to have DriversEduca-

tionAutomobile inherit from Automobile. By doing so, DriversEducationAutomobile will now automatically contain all of the variables and methods in Automobile. The only remaining task is to add those variables and methods specific to DriversEducationAutomobile.

Another benefit of using inheritance in this manner is that any changes made to Automobile will automatically be reflected in DriversEducationAutomobile. When an object inherits from another object in this way, the parent class is said to be the superclass and the child class is said to be the subclass.

Listing 2 provides an example. In this case, the brakeUsingInstructorsPedal() method of the DriversEducationAutomobile class performs some specialized behavior, then simply calls the brake() method that it inherits from Automobile. Note that in Java, inheritance is indicated by the use of the extends keyword.
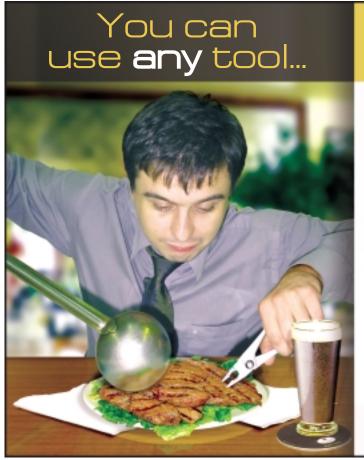
Subclasses can have other more interesting behaviors as well. A subclass can override the inherited behaviors of a superclass method, providing a specialized behavior that will automatically occur when the method of the subclass is called.

By using inheritance, developers can implement a hierarchical tree structure of classes by which specialized behaviors can be implemented at many different levels.

Encapsulation

Encapsulation revolves around the principle of exposing only as much information about an object as is absolutely necessary. In fact, there are actually two forms of encapsulation – "implementation hiding" and "information hiding."

Let's return to the automobile example to explain why encapsulation in its first form, implementation hiding, is so important. Suppose our automobile was manufactured with anti-lock brakes (ABS) and, in addition to the drive() method, our Automobile object also had brake() and triggerABS() methods. As many people know, the normal braking system of cars with ABS will determine when to

cause the ABS system to engage. We would assume that somewhere in the logic of the brake() method would be a call to the triggerABS() method. However, if we expose the triggerABS() method to the outside world, the effects may not be desirable. Suppose someone bypassed the brake() method and called the triggerABS() method directly. This could cause an accident. It's therefore desirable that only the logic of the brake() method call triggerABS() when it deems it appropriate. To prevent the above accident scenario from occurring, the concept of encapsulation dictates that the triggerABS() method not be exposed to any user of the class but only internally to the Automobile object.

In Listing 3, AutomobileWithABS extends Automobile and inherits Automobile's brake() method. Notice how AutomobileWithABS overrides the brake() method and adds its own customized version of brake(). If it's determined that it is necessary to engage the ABS system, then the triggerABS() method is called. If it is unnecessary, the brake() method of the Automobile class is called. Notice further that a triggerABS() method has been added that does not exist in Automobile. Note that the signature of this method is private, indicating that it can only be called internally to the AutomobileWithABS class.

The second form of encapsulation, information hiding, is equally important. Consider that our Automobile class could maintain a variable that contains the revolutions per minute (RPMs) at which the engine is running. This variable is set by the engine in reaction to various driving events: the Driver calling the accelerate() or brake() methods, for instance. What would happen if this variable were made available to be modified by the Driver class? At best the results would be unexpected. Since only the engine should set the RPM variable, it makes no sense to allow this variable to be altered by any external class.

In Listing 4, Automobile has an rpms variable declared as private, meaning that the variable cannot be accessed from outside the class. Two methods allow access to this variable, however. The getRPMs() method is declared as public and allows any other class to obtain the RPMs value. The setRPMs() method is declared as private, thus preventing any class other than Automobile from altering this value.

In short, encapsulation ensures that the inner-workings of an object are hidden from the outside world. Further, encapsulation ensures that both the hidden implementation (methods) and the hidden information (variables) of an object can be altered without negatively affecting other objects on which those objects depend.

Interfaces

In the Java programming language, an interface is a contract specifying which properties and behaviors an object will support. Interfaces allow the processing of objects based on a type rather than a class. (It's important to note that not all OO programming languages support the concept of interfaces.) Let's include our automobile again in an example. Suppose that a VehicleFueler class has been created. On an assembly line, two types of products are produced – automobiles and trucks – and each type of object (Automobile and Truck) gets instantiated somewhere in the assembly line process. Just before these products roll off the assembly line, both products receive a certain amount of a similar type of fuel from the VehicleFueler. The VehicleFueler class doesn't care whether the object is an automobile or a truck. Its only requirement is that the vehicles have some means of properly receiving fuel. In other words, the VehicleFueler requires a contract between itself and the two types of products that roll off the assembly line. A perfect solution to this problem

Listing 1
```java
public class Automobile {
 private String manufacturerName;

 public Automobile() {
 }

 public String getManufacturerName() {
  return manufacturerName;
 }

 public void setManufacturerName(String manufacturerName) {
 this. manufacturerName = manufacturerName;
}

public void drive() {
 … drive behavior …
}
}
```

Listing 2
```java
public class Automobile {

… other variables…

public void brake()
{
 … brake behavior …
}

… other methods …
}

public class DriversEducationAutomobile extends Automobile {
 public void brakeUsingInstructorsPedal()
 {
  … specialized Instructor braking behavior …
  brake();
 }
}
```

Listing 3
```java
public class AutomobileWithABS extends Automobile {
public void brake() {
  if (some condition warrants engaging the ABS system)
  {
    triggerABS();
```

```java
  }
  else
  {
   super.brake();
  }
}

 private void triggerABS() {
  … implementation to engage the ABS system …
 }

}
```

Listing 4
```java
public class Automobile {
 … other variables …

 private int rpms;

 … other methods …

 public int getRPMs() {
  return rpms;
 }

 private void setRPMs(int rpms) {
  this.rpms = rpms;
 }
}
```

Listing 5
```java
public class Automobile implements Fuelable
 {
  … other methods
  public void refuel(double amount)
  {
   … some implementation
  }
}

 public class Truck implements Fuelable
 {
  … other methods
  public void refuel(double amount)
  {
   … some implementation
  }
}
```

is the implementation of an interface. By implementing the following interface:

```
public interface Fuelable
{
  public void refuel(double amount);
}
```

the Automobile and Truck classes are agreeing to expose an implementation of the refuel() method. Automobile and Truck will look something like Listing 5.

VehicleFueler can now process either of the vehicles based on the type of object that it's processing rather than the object's class, as in the following:

```
Fuelable fuelable1 = new Automobile();
Fuelable fuelable2 = new Truck();
…
fuelable1.refuel(0.5);
fuelable2.refuel(0.5);
```

Another advantage of using interfaces in this way is that VehicleFueler does not need to concern itself with the implementations of either the Automobile or Truck refuel() methods. In fact, the details of what happens in the two objects' versions may be completely different and probably unknown to VehicleFueler. All VehicleFueler needs to know is that by implementing the Fuelable interface, Automobile and Truck encapsulate the functionality required for those objects to receive fuel.

Polymorphism

Polymorphism is the concept of an object changing its form to conform to a generic object abstraction. Varying definitions and implementations of polymorphism exist, most beyond the scope of this article. Yet the most common and easily usable definition is that of parametric polymorphism. Parametric polymorphism uses the concept of a common interface or lowest common denominator in terms of inheritance to implement a single abstraction across a number of types. To reuse the example, instances of Automobile or Truck can be inserted into a Collection object even though Automobile and Truck may be objects of a different class.

```
Automobile auto = new Automobile();
Truck truck = new Truck();

java.util.List list = new java.util.ArrayList();
list.add(auto);
list.add(truck);
```

The add() method of java.util.List (as well as many of the "add" methods of Collections classes) takes an instance of java.lang.Object as a parameter. Since Automobile and Truck are ultimate subclasses (see Inheritance discussion, above) of java.lang.Object, the code works. The root of the word polymorphism is morph, which means to change form. The example does essentially that as the added objects, regardless of class or type, are inserted into the Collection object due to their compliance with the inheritance requirements.

Conclusion

This article explained some of the core object principles in OO programming. Mastering these concepts will not only make you a better, more productive programmer, but will also make your code easier to write, easier to maintain, and more easily understood by others.

If you are just starting out, it may take time to integrate all of these concepts into your OO programming repertoire. But with patience and practice, you'll see yourself develop skills that will start to come to you automatically as you begin to think with "objects." ✐

Writing **Testable Systems**

Dan North

Complex enterprise applications are generally hard to maintain, and risky and difficult to change. As a new developer on a team, a large legacy code base is often difficult to understand, especially when the code has evolved over a long period and new functionality has been grafted onto an existing application. The situation is exacerbated by the usual time and delivery pressures of commercial enterprise systems development. With a modest amount of effort, however, it is possible to design and develop applications that are well architected and easy to change, and moreover that remain so as they grow and evolve.

What Is a Testable System?

There are certain characteristics that are common to all clean, well-architected applications. Features such as modular design, good separation of concerns, loose coupling, and high cohesion – behavior in one and only one place within an application – are all well-established best practices. Writing applications with testability in mind tends to lead to systems that display these characteristics. For instance, focusing on the ease of black- and white-box testing of individual modules is a sure-fire way to generate cleanly separated functional units. Saying that a module is easy to unit test is the same as saying we know exactly what the module should do, under

application or maintaining an existing enterprise application. I'll offer pragmatic advice and techniques for choosing – and sticking to – clean and extensible architectures, and help you contrast different approaches effectively from a testability perspective.

What Is an Enterprise Application?

If you ask 10 developers what constitutes an enterprise application, you'll most likely get 10 different responses, but in general an enterprise application will display some, if not all, of the following characteristics:
- Integration with existing legacy applications (often in different languages and on heterogeneous hardware and operating systems)
- Some sort of persistence, typically a relational database
- Multiple tiers, with at least presentation and business logic, and often data access broken out into separate tiers
- Many users, across many widely distributed sites, often in different countries
- High throughput and performance requirements
- High availability and resilience

It is these sorts of environments I will be discussing in this series. I'll look at various "typical" Java application models, such as Web apps, full-blown multitier J2EE appli-

> " This new column looks at what makes an application testable, and provides practical help and real-world advice for writing applications in a clean and test-driven fashion"

what circumstances, and that we know (and can prove) its boundary behavior. Saying that a module is easy to integration test implies that it is easy to deploy and configure.

To keep ourselves honest and consistent, we write tests as we go to define and prove that components work in a particular way, and we automatically run all those tests often. As discussed in my previous article ("Test-Driven Development Is Not About Testing" [*JDJ*, Vol. 8, issue 11]), each test case becomes a model client of the module it is testing. This can be true at any level of granularity, whether the module is an individual class, a deployed component such as an EJB or servlet, a service, or an entire subsystem. The size of the unit may vary, but the purpose of the test case remains consistent.

This new column looks at what makes an application testable, and provides practical help and real-world advice for writing applications in a clean and test-driven fashion. Its focus will be on simplicity of design, from the viewpoint of a developer wanting to work with clean code, not as a professional tester. The target audience is enterprise application developers who are either embarking on a new

cations, and event-driven architectures, and explore how strategies such as MVC and inversion of control can improve their design and structure. In addition to the typical *n*-tier enterprise architecture, we will look at message-driven applications, Jini, JavaSpaces, and various other nonclient/server models for distributed application development.

Why Should It Matter to Me?

For notoriously hard-to-test components such as Swing widgets, EJBs, servlets, and database connections, we'll look at ways to test outside of the container and review tools and frameworks that help promote loose coupling and overall application goodness. I hope you'll find this new column pragmatic and directly applicable to your development world.

The first article in the series will present some alternatives to Pet Store–shaped enterprise applications, and subsequent articles will look at the testability of these architectures and enable you to create and evolve clean, maintainable enterprise systems using them. 🖉

Dan North has been writing software for 12 years, and is a programmer and coach for ThoughtWorks (www.thoughtworks.com), a software development consultancy, where he encourages people to write tests.

dan.north@thoughtworks.com

isavix.net

Searchable Books

Search electronic versions of over 1,500 technical books

- Conduct searches across 1500+ technical books from publishers such as Addison Wesley, Microsoft Press, O'Reilly, Prentice Hall, and many more. Zero in on answers to time-critical questions instantly.

- Read the books on your Bookshelf from cover to cover. Or, simply flip to the page you need.

- Browse books by category. Researching any topic is a snap. From XML, to database to .Net, you'll find your answer here.

eLearning Courses

The Isavix High-Impact eLearning Series™ is designed to deliver online training in the most efficient manner possible. Each course is approximately one hour in length with no "homework" assignments and can be viewed in Flash and PDF formats. Sample courses include The Struts Framework, Introduction to Apache ANT, ASP.NET Crash Course, and more.

Certification Exam Preparation

Whizlabs exam simulators are available for Sun, IBM, BEA, CISCO, Microsoft, Oracle, CompTIA, CIW and Red Hat certifications. These products have been developed by experts who are considered the authorities in their subjects.

From computer professionals to Certification Training centers to College and University students, to hundreds of corporations ... the Whizlabs solutions have helped over 150,000 people get their certifications.

FREE Resources and Tools

isavix.net is a site that provides eLearning, information sharing, software exchange, helpful tools, and valuable links for emerging technology (e.g. J2EE, .NET) professionals.

Our free tools and resources include Discussion Forums, Code Exchange, My Secure Files, Quick Polls, My Bookmarks, Download Links, My UML Diagrams, My eLearning, and much more.

Custom Community Portal
For Your Organization

If you like the many benefits isavix.net provides but would want a custom version of our portal available exclusively for your organization, we can do that for you!

Our community can either be installed within your organization's Intranet or provided to you as an ASP solution.

The benefits include an enhanced user experience, professional development, improved team dynamics, effective project management, and better resource allocation. The ROI for an organization is high because developers can save from a few minutes to hours a day since the relevant information is readily available to them, when they need it!

Corporate and Government solutions available ... contact us for details.

isavix
Delivering Emerging IT Solutions

Isavix Corporation
Herndon, VA
1-866-472-8849
info@isavix.com

An Interview with
Richard Monson-Haefel

Interviewed by
Alan Williamson

JDJ's Alan Williamson caught up with the well-known Java luminary Richard Monson-Haefel just after his recent appointment to the JCP Executive Committee. Alan asked Richard a broad range of questions including his view on how Java as a language has matured and what we are likely to see in the next year.

JDJ: Thank you, Richard, for taking the time to talk to JDJ. I believe congratulations are in order: election to the JCP Executive Committee. How do you feel about that? Did you have to lobby for support?

RMH: I'm honored to serve on the Executive Committee, and I didn't do any lobbying for it. I think some of the JCP members figured another individual (as opposed to a corporation) might add something to the process – Doug Lea has served as an individual on the EC for a couple years now and was voted back in by a landslide.

There are two Executive Committees (ECs): one for the J2ME platform and one for the J2SE and J2EE platforms. I will be serving on the J2SE/J2EE EC, which means I'll be working with other EC members to determine which JSRs are approved for development and become final specifications. That's a huge responsibility. There are only 16 members on the EC and you need a majority with at least five "yes" votes in order for a JSR to pass each milestone in the JCP process. That means as an individual, I'll have a lot of responsibility and potentially a large impact on how the Java platform evolves for the next couple of years. I don't take that lightly.

JDJ: What does this mean for you? What are your duties?

RMH: I'll be flying to California and elsewhere about four times a year for in-person EC meetings, and also meeting in conference calls about once a month. It's a lot to take on, but the opportunity to help shape the future of Java is just too important to pass up.

I'm an independent, which means I won't be representing a big business interest. I'm going to do my best to represent Java developers who are not a part of the process. I also want to see that the needs of the open source community are represented – Apache already has a member on the EC, Geir Magnusson Jr., so I feel I have an ally in that endeavor. I'll also be looking to Doug Lea for guidance on how to work effectively as an independent in an organization dominated by big business interests.

JDJ: Sounds like a tremendous amount of responsibility. Is there a direction you are looking to steer Java? What areas will you be specifically looking at when new proposals come through?

RMH: I'll be looking for a couple of things, including JSR overlap, industry relevance, and ease of use. The first two criteria are part of the charter of the EC members. The EC wants to avoid overlap among JSRs and also to ensure that the JSRs are useful and needed by their target industries. The ease of use thing is a personal crusade. Java is an awesome platform and J2EE is quite impressive, but the number of APIs is massive and their complexity can be daunting. I'll be looking for JSRs that improve usability from a developer's perspective.

JDJ: The JCP gets a fair amount of stick from various quarters of the Java community. On the whole do you think it's working?

RMH: The JCP has, in my opinion, improved a lot with the latest version of the process. There is always room for improvement, but overall I think the JCP has done well. I wouldn't have taken my seat on the EC if I felt otherwise.

JDJ: You have been writing about Java for many years now. Over that period, how do you feel we've matured as a community?

RMH: In the early years everything revolved around the Java programming language. How to use the core APIs and what VM technologies are best. Those things are still important, but eventually developers turned their attention to APIs outside of the core, like JDBC and EJB – and now J2EE and J2ME. Lately developer interest has turned to architecture and design – a good sign that Java has matured. Rather than focusing on how to use Java APIs, the community is turning its sights on when and where to use Java in larger enterprise systems.

Another thing that changed is the question of Java itself. When I first got started in Java (end of 1995) I spent half my time programming in the language and the other half evangelizing it. Today it's very different. Java is the first programming language taught by many learning institutions and just about every large corporation is using it somewhere – often in mission-critical applications. The question has changed from "Why should we use Java?" to "Which Java technologies should we use?"

JDJ: Tell us a little about your latest book, J2EE Web Services. Are Web services finally being adopted in the field?

RMH: This is probably the best book project I've worked on. It took me twice as long to write as I originally thought it would because it turns out that J2EE Web services is a huge topic, and its takes effort to explain it clearly. I (and a lot of other people) think Web services are important because they're really useful when connecting disparate systems. This isn't necessarily because Web services is based on XML and HTTP; it's because Web services is endorsed and supported by every major software vendor, including Sun,

Microsoft, IBM, BEA, HP, and Oracle. The cover of my new book has an 18th-century Japanese woodblock print of a giant wave, which was inspired by the tidal wave of industry support behind Web services. It reminds me of the kind of support we saw behind Java and XML when they were first introduced.

The book I wrote is really a developer's guide to the Web services technologies (XML, SOAP, WSDL, and UDDI) and the J2EE Web services APIs (JAX-RPC, SAAJ, JAXR, and JAXP). Although I originally planned to cover architecture and design, I had to stick to explaining the technologies because they are so vast and deep. In the end, I'm enormously satisfied with the book – I believe it will be very useful to developers implementing J2EE Web services.

JDJ: Are you working on anything at the moment? Are we going to see another book from you in 2004?

RMH: O'Reilly will be publishing *Enterprise JavaBeans, 4th Edition* in 2004. I'm working on that right now and I'm almost finished.

As an author, my major focus for 2004 and 2005 will be on a new book about the Java platform, tentatively titled *This Is Java*. There's a lot of books that focus on the anatomy of Java, its programming syntax and core APIs. This book doesn't cover that stuff. Instead, *This Is Java* will address the physiology of the Java platform by explaining how Java manages threads, I/O, networking, classloading, and things like that. Although it sounds like a JVM book, its scope is broader. The book not only explains how Java works under the hood, but why it was designed the way it is, and how to leverage that design to produce more efficient and performant code. The objective of this book is to make every reader an expert on the Java platform.

I think today most Java developers are like race car drivers who know how to drive, but don't understand how their car is put together. In the sport of auto racing the best drivers know their cars inside and out and can build, tune, and fix them as well as drive them. Similarly, the best Java developers know how Java works inside and out. *This Is Java* will make that knowledge accessible to all Java developers.

JDJ: Take a moment and have a think about Java's history. Name one thing that Java has done right and, conversely, is there anything in Java's history you would like to change?

RMH: Java was built from the ground up to be an object-oriented, network-savvy programming language. I think we got that right. James Gosling and crew created an elegant and powerful programming language. The thing I look back on with less excitement is Swing. I think Swing set the Java platform off its desktop course and gave ground to other technologies. I'm also a bit remorseful about the complexity of the J2EE APIs – while they are pretty cool, they are on the whole, I think, too complex and probably a bit daunting for many developers.

JDJ: Have we lost the desktop battle to the likes of Flash and .NET?

RMH: We have lost a couple battles but not the war (sorry for the cliché). In my opinion, AWT delegated too much to the OS while Swing delegated too little – Swing is also too complex. A compromise, perhaps something along the lines of SWT, is needed to make Java as popular on the desktop as it is on the server. I'm confident that we will get there; it's just a matter of time.

JDJ: Let's imagine for a moment I wanted to invest serious money in a Java startup company. What type of company would you advise me to put my hard-earned dollars into? What's going to be big in the next 18 months?

RMH: On the server side, I think tools that enable developers to quickly and easily expose existing and new systems as Web services will be big in 2004. On the desktop, I would keep an eye on the Java IDE market. I think we'll see a lot of consolidation in that area.

In the long haul, I think open source is going to pressure vendors in many markets to focus on usability rather than utility. Open source has a tendency to commoditize markets, but it also offers vendors a lot of opportunity when it comes to making utilitarian software easier to manage and use.

Of course, these are only my opinions, and if I was right more than half the time I would probably be retired already. I'm not an investment advisor, so take these opinions with the appropriate amount of salt – a metric ton or two should do.

JDJ: Many have complained J2EE is too unwieldy and complicated. Do they have a point? Or is the latest generation of development tools easing the pain?

RMH: Yes, they do. For the past few years the J2EE community has focused on supporting enterprise computing in a way that is portable and, more recently, interoperable across a host of products. That's pretty difficult to do without introducing some complexity because enterprise systems are inherently complex. However, now that we have addressed the complexity to a large degree, I believe it's time to turn our attention back to ease of use. Specifically, I'm interested in offering application developers a simpler programming model for the entire J2EE platform. I don't want to throw out the APIs we have today, because they offer a level of granularity and portability that's crucial to the J2EE platform, but we should offer a broader abstraction of the J2EE platform that will rest on top of the standard APIs, a programming model that is more implicit and intuitive and less complex and explicit. That is going to be a big focus for me over the next couple of years – making J2EE easier to use without sacrificing the portability or sophistication of the platform.

> "Lately developer interest has turned to architecture and design – a good sign that Java has matured. Rather than focusing on how to use Java APIs, the community is turning its sights on when and where to use Java in larger enterprise systems"

Joe Winchester
Desktop Java Editor

To Layout()
or Not to Layout()

A problem encountered by any GUI is – if the user resizes the application window at runtime, how should this be handled? The most desirable effect is that the controls flow into the new space to make the best use of it (lists and trees grow while buttons remain a fixed size), and an overall reduction in size doesn't clip or lose information.

Using layout managers, this problem is bread and butter for Java. The favored choice among Swing programmers is GridBagLayout where each component is placed in an imaginary grid cell. The height of each row or width of each column is made large enough to accommodate its largest component, and every other component can specify rules as to how it anchors itself to the cell's edges together with any padding, insets, and how may cells or rows it should span, as well as a weighting for how excess space in the overall GUI is divided between its rows and columns.

For some Java developers, learning the nuances of GridBagLayout is a rite of passage easily mastered, while for others it's avoided altogether. I know programmers whose policy is to ignore layout managers entirely and, instead, hard code the coordinates of a GUI's components with

this is a very elegant solution, it still requires that you have an understanding of how GridBagLayout works.

It's easy to blame the GUI builders for why layout managers are hard, but they're only working with what's available in the JFC; the original programmers who designed GridBagLayout came up with a good engineering solution to a difficult problem.

What about tackling the problem slightly differently, with the layout logic concerned only with rearranging children when size is altered? Language locales could externalize a completely redesigned GUI class for that country's users. For resizing, there's a general set of rules that could be followed: tables and lists want to grow to occupy new space, buttons and labels stay the same size, text fields can be made wider but not deeper (unless they're multiline text fields), and so forth. The default algorithm might work okay in the 80/20 case, and if further customization were needed this could be achieved by specifying constraints that overrode the defaults.

Having null layout or GridBagLayout is all or nothing at the moment, whereas this would be a shade of gray in between. The developer experience could be that compo-

> " What I hope is that we as Java developers can continue to
> ## think of new ways to innovate and improve the language and tools"

setBounds(Rectangle bounds). Because null layout fixes every component's bounds at design time and translated user-visible strings change length at runtime, applications then have to standardize on a single language and explicitly prevent users from resizing windows with setResizeable(false).

One reason this occurs is because developers often use tools to help build their GUI, and there's something very compelling about being able to size and position controls in null layout. The controls go where you tell them. It's a kind of WYPIWTG or "Where you put is where they go." In most Java GUI builders, with anything other than null layout, the experience is closer to "Put it down and watch everything move." It's rather like a car where the driver steers the wheel to where he wants to go and all the roads, buildings, and trees move and change size at the same time. The driver makes another correction only to find he can't go there and instead has to use dashboard buttons to change some of the car's properties before altering course. While this experience may be okay for bicycle-riding cartoon clowns, it's not one that Java GUI developers should suffer unnecessarily. Some tools recommend doing everything in null layout (where precise positioning of components can occur) before switching the container to GridBagLayout, whereupon an algorithm kicks in that generates a guestimate GridBagConstraint. Other tools use an alternative view of the container where you manipulate components in an explicit grid and, although

nents are positioned and sized as with null layout, so GUI elements remain where they're put at design time, with the constraints being an additional parameter for what the growth rule is. A colleague recently pointed me to the FormBuilder package from http://www.jgoodies.com that goes some way toward providing this experience with a simple programming API, although I'm not yet aware of its integration in any GUI builders.

In a past life I programmed Windows GUIs and we had a rubberband.vbx, which is analogous to a custom container with all the smarts for resizing logic built in; however, control placement was done on an x and y grid. Another idea once suggested to me was that a GUI builder should allow you to design at two resolutions, and then extrapolate all the intermediate screen sizes.

Whatever the solution might be, what I hope is that we as Java developers can continue to think of new ways to innovate and improve the language and tools. If anyone hasn't already seen it, please visit javadesktop.org, which is part of the java.net community and is a great way to engage the Swing developers in a nice "flame-free" forum, as well as see what's in the pipeline. The other people to talk to about how to make user interfaces better are…your users. Talk to them, listen to them, and push their feedback down the stack so we can all work to keep improving and enjoying Java clients. ✐
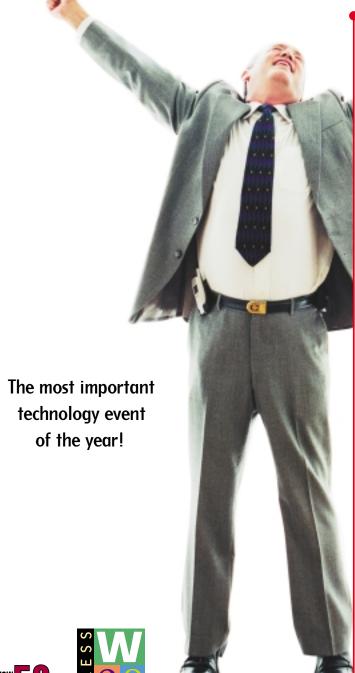
Joe Winchester is a software developer working on WebSphere development tools for IBM in Hursley, UK.

joewinchester@sys-con.com

DESKTOP

CORE

ENTERPRISE

HOME

Multiplatform Frames

INTEGRATING INTO SWING-BASED APPLICATIONS

by Todd Brett

Java is enjoying a renaissance on the desktop, yet developers are still frequently required to invent methods for dealing with application management tasks within J2SE. One common problem involves the design decision regarding whether an application should adopt a Single Document Interface (SDI) or Multiple Document Interface (MDI) window presentation. This article discusses the issues surrounding window frames and their environments when using Swing.

SDI vs MDI Presentations

The two class models that support SDI versus MDI presentation environments within J2SE are incompatible with each other for event processing and setup, which makes their consideration important for new applications. Designers cannot push this problem to a later life-cycle phase or product revision because an application's data presentation should cater to the user's expected work environment. This is true for users of Mac or Windows programs who have different expectations for how their GUI will appear and respond. If a developer must make a design compromise, then the application's usability or appeal beyond a select platform will be limited. Careful organization of application logic allows for an approach that supports a dual SDI and MDI presentation within the same application in the context of the J2SE 1.3 and 1.4 Swing libraries.

A Common Frame

The most basic presentation structure available to Swing-based applications is javax.swing.JFrame. JFrame implements a SDI window that presents a user interface by hosting customized javax.swing.JComponent objects. MDI windows are provided to developers through javax.swing.JInternalFrame, although these structures are not actually native windows; rather, JInternalFrame implements a JComponent that understands how to draw itself with a title bar and behave as an embedded frame.

The event models used to track button clicks and other events within these window structures are incompatible as a result of JFrame and JInternalFrame's dissimilar inheritance trees (JFrame is a true java.awt.Window derivative, whereas the latter is an embedded window element). Fortunately, an application's data presentation layout and its associated business logic connections

are provided by JComponents supplied to each frame's content pane. Further, the designers of JFrame and JInternalFrame were kind enough to give many of the important initialization and manipulation methods similar signatures (parameter lists and return types). This combination of shared behavior and structure allows developers to apply the Adapter design pattern by declaring a representative interface that includes all of the methods required for manipulating a typical frame with the added safety of deferring the definition of the application's interface to some JComponent object. The nature of Java's inheritance rules makes possible the use of this strategy: any class can extend one other class while simultaneously implementing any number of interfaces. Figure 1 provides an example of how an interface can represent both a JFrame and a JInternalFrame simultaneously.

Similar tasks do not necessarily use methods with the same signature, and some of the functionality in one frame class is not present in the other class due to specialization. Choosing either class as the primary template for the representative interface easily solves these behavioral discrepancies, leaving the alternate derived class able to map missing functionality using a combination of method calls for native frame logic. The result of this logic combination is an implementation that is familiar to Swing developers yet also compatible with a traditional three-tier application design where presentation, business logic, and IO logic layers are kept distinct (see Listing 1).

This common frame foundation does not completely provide the desired dual presentation application, however. Although the interface combines access to both SDI and MDI frame implementations, an application must also concern itself with window creation and management. SDI and MDI presentation components must allow for their seamless exchange at runtime in order to accommodate different platforms. Further, JInternalFrame objects cannot operate without a managing JDesktopPane instance.
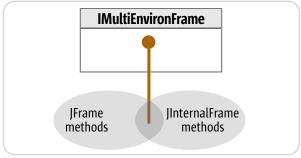

Figure 1 A common frame interface

Todd Brett has over 10 years of experience architecting and implementing well-behaved, multiplatform applications in a variety of languages. He holds an MS with an emphasis in object-oriented technologies and technology management from Regis University in Denver.

toddbrett@insight.rr.com

The Frame Environment

The Abstract Factory and Factory Method patterns provide excellent design entry points for the extension of the developing framework by defining component relationships to ease frame creation and management for the application. As before, using interfaces allows the framework to substitute environments, resulting in an entire change of the system's presentation strategy with no effect on surrounding logic.

SDI presentation requires the least complex environment because JFrame provides an independent presentation that does not require additional components to enable its display. An MDI environment is more complex because an active JDesktopPane object embedded within a master JFrame is required to enable the rendering and manipulation functionality required by JInternalFrame objects. Apart from these differences, the two environments' needs are essentially the same. Architects can therefore consider two design strategies when creating the concrete classes, either of which works, providing the application will operate within the Swing framework:

- The SDI environment can be defined and then a MDI environment can be derived from this, recognizing that MDI is merely a complex SDI presentation.
- An abstract base class from which both the SDI and MDI frames inherit can be defined.

Our dual environment framework uses an abstract class to provide the foundation for each of the concrete environments, as shown in Figure 2. This superclass provides concrete derivations with a collection to track each environment's frames, as well as shared functions for general frame maintenance. Use of AbstractList for the frame collection type gives applications the advantage of threadsafe behavior through the assignment of Vector objects, while faster performance can be accomplished by substituting ArrayList instances instead. Platform-specific presentations, such as the proper positioning of the Mac OS's global menu bar, can be effectively defined at this concrete level for general application use.

The derived SDI environment's need for only the addition of creation logic to handle SDI frame instantiation makes this component's simplicity apparent. The MDI environment needs to go a step further by providing initialization routines for a master application JFrame and an associated JDesktopPane through which the environment will display the internal frames to users. Developers should be able to define environments for technologies other than Swing in a similar fashion.

The Application Environment

In addition to hosting frames, the system should allow for tasks such as tiling windows and registering predefined frame objects that the application can clone in order to provide common document presentations. For this purpose, the framework defines a frame management interface and abstract base class.

The interface component provides access to the session environment and any existing frames. A middleman frame creation method is provided for both the application's convenience and the opportunity for subclasses to clone and return commonly requested window types, such as a document workspace previously registered with the manager and its environment. The abstract component houses the presentation environment for each new session and can also be defined to select the appropriate environment early in an application's execution. Figure 3 illustrates the overall architecture by which the application can associate a presentation system with the other essential functionality.

Platform-specific rules not explicitly handled by the selected environment are best defined at the management-component level. More complex applications, such as a document-

centric system (e.g., a multimedia word processor), should build upon the abstract class to provide the logic for frame interaction and arrangement from the user perspective. This includes business rules for activities such as window layout, layering, and positioning. Developers can extend the management system through the addition of custom window behavior logic and event listeners to provide the application with its own global frame management architecture.

Menu Arrangements

The framework now provides for SDI and MDI windows, their host environments, and a general application manager. Locations exist to add platform-specific logic as required, and environments can be swapped out for one another without any detrimental impact on the program's logic. A remaining requirement is to support menu arrangement, which is different for each presentation environment.

Menus displayed within an SDI application appear in the same menu bar for any given window, regardless of whether the menu relates to the application, the document, or another conceptual division of the user's workspace. By comparison, MDI applications display their global application and help menus in the master window's menu bar, while project and document menus exist within local menu bars of the internal windows. The presentation framework can accommodate these divisions by creating specialized menu managers and containers that provide each presentation environment with all of the required information to ensure the appropriate arrangement of divisions.

An understanding of the existing Swing menu framework is required to properly implement these menu components. A JFrame uses a JMenuBar object to host the window's various menus, which are each represented by JMenu instances. A JMenu can contain multiple menu items in the form of JMenuItem objects, as well as submenus represented by additional JMenu objects. A JFrame's setJMenuBar method is executed with the
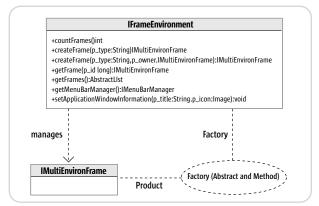

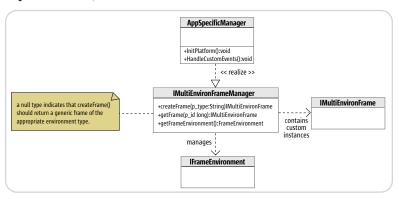**Figure 2** The relationship between frames and environments

Figure 3 The foundation for application-specific window manipulation

desired primary JMenuBar instance as the parameter. Any JMenu object will only draw itself in one visible JMenuBar instance at any given time. This means scenarios in which a menu bar would logically apply to several different frames (such as document windows of the same type) require the cloning of the menu bar and its contents. Figure 4 shows how this can be achieved using a template manager and a customizable menu group interface.

The menu manager is aware of associated menu groups and can provide limited manipulation operations on those groups to affect attributes such as visibility. In addition, the manager also understands a predefined arrangement of menu group types that guide the display of associated groups from left to right in the order of their significance to the application and to the user. This ensures a consistent menu arrangement regardless of which frame is being displayed. In a simple example these arrangement values are predefined, but a more powerful type registration mechanism with arrangement-weighting schemes could be supplied for systems requiring extensive flexibility.

Extending JMenuBar provides the application with a Swing-aware management component that ties directly into the framework's presentation architecture. Implementing the Clonable interface allows the clone() method to be used by the presentation environment to produce clean copies of the menu bar and its menus.

The concrete SDI environment clones the manager and distributes this to newly created frames to provide the application global menus, but defers the addition of document-specific menu groups to each frame's custom JComponent. The benefit of this arrangement is that application and help-level menu groups display in consistent arrangements with the same content across different windows. The concrete MDI environment directly associates the menu bar manager only with the application's master frame, but still provides copies of the manager to internal frames to allow the opportunity to provide a consistent project menu group set across all document frames. An alternative approach would involve documents supplying the same functionality through custom JComponent objects associated with internal frames to provide project menu groups alongside document-level menu groups.

A menu group is aware of its associated JMenu objects, but otherwise acts as an environment setup and container class. While visibility operations exist, such as toggling enabled states for all associated menus, the definition of status operations for each JMenu is more generally left to the application. Programmers benefit from the menu group's ability to create convenient templates for creation and initialization, such as an Edit menu for a text editor or the File menu for an application. New menu groups
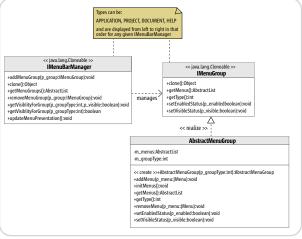
can then be quickly created and attached to a new window's menu bar manager with a minimum of code (see Listing 2).

Conclusion

J2SE provides much of the functionality necessary to create applications for all of the major platforms, but multiplatform presentation has always been a problem. The goal is to create a single framework that represents common methods for different frame types with the ability to host frame instances. This allows developers to focus on business logic while remaining comfortable with the knowledge that their application will be well behaved across all of their customers' computing environments.

This article walked through the design and key implementation considerations for such a framework that can be easily integrated into existing Swing-based applications. A complete implementation of this framework can be found at the author's Web site, http://home.insight.rr.com/thebretts/todd/research/. ✎

Resources and Further Reading

- Apple Computer, Inc. (2001). *Aqua Human Interface Guidelines. Inside Mac OS X.* Apple Computer, Inc.
- Gamma, Erich, et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Sun Microsystems, Inc. Java 2 Platform, Standard Edition, v 1.4.2 API Specification: http://java.sun.com/j2se/ 1.4.2/ docs/api
- Sun Microsystems. (2001). *Java Look and Feel Design Guidelines, Second Edition*. Addison-Wesley.

Listing 1

```
1   class SDIFrame
2          extends JFrame
3          implements IMultiEnvironFrame
4   {
5     // interface methods do not need
6     // to be specified--JFrame
7     // provides these.
8   }

1   class MDIFrame
2          extends JInternalFrame
3          implements IMultiEnvironFrame
4   {
5     // Interface methods specific to
6     // JFrame are defined using
7     // JInternalFrame method mappings.
8
9     public void setIconImage(
        Image p_icon )
10    {
11      setFrameIcon(
          new ImageIcon( p_icon ) );
13    }
14
15    // ...
16  }
```

Listing 2

```
1   class ApplicationMenuGroup extends
    AbstractMenuGroup
2   {
3     // ...
4
5     // Very simple implementation.
6     public Object clone()
7     {
8       ApplicationMenuGroup newClone =
          new ApplicationMenuGroup();
          // Could also use copy
            constructor.
9       newClone.initMenus();
10      return newClone;
11    }
12
13    protected void initMenus()
14    {
15      JMenu fileMenu =
          new JMenu("File");
17      addMenu( fileMenu );
18      // ...
19    }
20  }
```


Figure 4 Menu arrangement components

If the Fallacies pervade networking, Sun was perhaps ideally placed to identify and surmount them, says Gosling. "We have a very strong engineering culture," he says. "A lot of our work is about building solid reliable systems."

Beginning with Java, Sun has explicitly grappled with the Fallacies. Java's "write once, run anywhere" approach celebrates heterogeneity, according to Gosling. While nobody exactly tacked the Fallacies up on the wall when Jini was designed, says Gingell, "If you look at Jini, it's apparent that what it's trying to do is confront some of the Fallacies, because the Jini team and Peter shared a view of those problems." Jini lets diverse devices discover and interact with each other without any administrator at all; it also can deal with multiple administrators acting divergently.

Indeed. "Almost everything in Jini is about dealing with that list," Gosling says. "It's all about the dynamic, spontaneous reconfiguration of networks in the face of a dynamic environment. Things getting plugged in, things getting broken."

The Liberty Project is another example of Sun engineering's acknowledgment of the single administrator fallacy, according to Juan Carlos Soto, product marketing group manager for Project JXTA and the jxta.org open source community manager. "Liberty provides a federated identity mechanism that makes it convenient for businesses to interact while still respecting privacy, because there's no single point of control," he says.
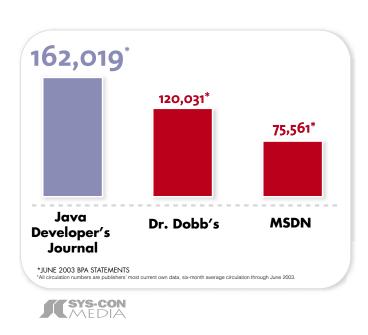
JXTA goes perhaps furthest in turning all of the Fallacies into truths. It creates an ad hoc virtual private network among peer devices that aren't always at stable addresses, Soto says, so it solves the problems of multiple administrators, changing topology, and heterogeneous networks. Moreover, because JXTA makes it easy to create multiple instances of a service, and peers cooperate to move messages through the network, it creates great resiliency that counteracts the inherent unreliability of networks. It's designed to be efficient and conserve bandwidth as much as possible, reducing the impact of Fallacies Two and Three, and it accepts every authentication method for flexible and tight security.

"P2P is turning the computer into both a server and a client," Soto says. "It's way beyond a single administrator and a homogenous network. It abstracts a lot of those issues away."

The Sun days are a long time ago for Deutsch, who now, as president of Aladdin, consults for technology and venture capital companies. He's a bit bemused that his short list of big goofs has become institutionalized as Deutsch's Fallacies. "If you had told me when I was at Sun that 10 years later, this one page of maxims was going to be one of the things I was best known for," he says, "I would have been floored."

Although networking has changed plenty since 1991, and some of the Fallacies, such as betting on a secure network, are more obvious, they are all still as applicable, Gingell says, and they continue to be driven into the hearts and minds of Sun's engineers. "Cultural reeducation is more important than any one product," he says. "If all our engineers understand networking at a very large scale…the products will take care of themselves."

The Rich Client
Strikes Back

Philip Brittan

Microsoft is redefining the application interface around rich clients, and if Java does not have an answer, it faces being cut off from end users. The answer lies in matching Microsoft's richness while trumping it on security.

There is a movement afoot in the world of enterprise applications toward "rich clients," reversing the past few years' trend that abandoned client/server in favor of Web-based interfaces. One of the most vocal champions of this is Microsoft, which would like to see a return to fat-client software bound to its Windows desktop operating system.

The customers' desire for rich clients is well founded.The deployment ease of HTML into ubiquitous browsers seemed a boon to CIOs, but as end users soon discovered, Web pages are not necessarily well suited to be interactive user interfaces. Studies have shown that poor usability has very significant economic consequences for enterprises (www.useit.com/alertbox/20021111. html). While the switch to HTML may originally have eased deployment issues for developers, users experienced decreased productivity and overall frustration with a poorer application. By contrast, rich client applications use native GUI components and are more interactive and

Start–like deployment mechanism. For many companies that tried to make Web-based front ends a standard for UI development, their users pushed back on functionality, leaving them to cast around for rich client alternatives. Java has so far failed to offer a viable rich client solution, and the danger is that these tech decision makers may turn to .NET and the allure of a native-style Windows client.

If the Java world does not offer a rich client alternative to .NET, it will be cut off from users and then corralled into a smaller and smaller sphere of influence. However, simply copying Microsoft and trying to offer a client-side programming model won't necessarily win favor. It's very hard for a technology to ever recover from a poorly received initial release, and despite the much-touted advances of previous releases I find that Swing has already been largely discredited among customers. If Java tries to simply copycat .NET, however, there will not be sufficient incentive to provide compelling differentiation for users (aside from cross-platform compatibility, which is often largely irrelevant in most enterprises).

One answer lies in changing the rules of the game. Microsoft is confounding the rich client user experience with a fat client architecture; however, these are two

> " IT managers and CIOs should be reassured that Java hasn't given up on traditional client/server topologies"

Philip Brittan is the founder and chairman of Droplets (www.droplets.com), a developer of rich thin-client technology for corporate network applications. Previously, he founded and ran Spheresoft, a software development firm; prior to that he was lead developer and CEO of Astrogamma, a financial software firm.

pbrittan@droplets.com

responsive, allowing users to work faster with fewer human errors, as well as for more sustained periods. True rich client applications (unlike "rich browser" applications such as DHTML and Flash) can also integrate with the desktop environment.

Although Java had its beginnings in client-side software, it failed to take root and has now largely retreated to the server, where it has built a veritable stronghold through J2EE. However, it's a stronghold that Microsoft is determined to crack so that it can grow its server market share. Microsoft's main weapon in this siege is .NET and the attack is two-staged: first, conquer the desktop with .NET by leveraging desktop market dominance with the focus being a compelling user experience; second, leverage private synergies between the client and server parts of the .NET Framework to drive from the client back into the server room.

The first part of this campaign is underway. Microsoft is pushing its rich "smart client" model aggressively, touting high usability, desktop/Office integration, and a Java Web

independent concepts. CIOs originally ran to a thin-client model of the Web because it offered real deployment, administration, and security benefits. The Microsoft Smart Client model has come a long way to try to address them, but still falls far short, especially with security – their perennial Achilles' heel.

Security is extremely relevant with the frequency and destructive power of malicious code attacks steadily increasing. A thin-client model is significantly more robust against viruses and hacker Trojan horses than fat-client alternatives. Microsoft's strategic dependence on a fat-client architecture means that they will likely never be able to address this problem completely, which leaves a very large opening for Java technology to offer a secure alternative to .NET. IT managers and CIOs should be reassured that Java hasn't given up on traditional client/server topologies, and we should be actively working to innovate in this area so Java developers can deliver high-end, rich client applications to their end users. ✐

edge 2004 EAST

DEVELOPMENT TECHNOLOGIES EXCHANGE

February 24-26, 2004

Hynes Convention Center, Boston, MA

ARCHITECTING JAVA, .NET, WEB SERVICES, MX, XML, AND OPEN SOURCE

JOIN THE EXCHANGE!

Become a part of the Development Technologies Exchange in Boston this February. Learn from the leading voices in the i-Technology world, see the latest developments from the companies that push the envelope of innovation, and network with the movers and shakers of IT.

The Edge 2004 East Conference Program holds the information you need to excel in today's competitive environment.

2

Features & Attractions

- 3 Days Packed with Education and Training
- Keynotes & Panel Discussions from Industry Leaders
- FREE Web Services Security Tutorial
- FREE .NET Tutorial "The Smart Client Perspective"
- Industry-Leading Certification Programs
- "Birds of a Feather" Discussions
- Round Table Discussions
- Opening Day Welcome Reception
- Compelling Case Studies & Best Practices
- Hands-On Labs
- Featured Product Demonstrations
- Exhibit Floor Featuring More than 40 Companies and Hundreds of Products
- Real-time SYS-CON Radio Interviews

Who Should Attend

- Software Developers
- Software Engineers
- CTOs
- CIOs
- Development Managers
- Application Developers
- IT Directors
- Technical Directors
- Analysts
- Consultants
- Programmers
- IT Managers
- Technical Architects
- Team Leaders
- Software Consultants

Register By JANUARY 24, 2004
SAVE Up To $300

3

Keynotes

Chris Hjelm
CHIEF TECHNOLOGY OFFICER, ORBITZ
Tuesday, February 24, 2004 – 10:00 a.m.

ORBITZ

Chris Hjelm is chief technology officer for Orbitz and leads the company's technology strategies, engineering, site architecture, and operations. In this role, he is responsible for all aspects of technology, ensuring that Orbitz continues its industry-leading role in providing the lowest cost and most consumer-friendly technology platform in online travel. Before joining Orbitz, Hjelm served as SVP for technology at eBay Inc., overseeing all aspects of its technology. Under his leadership, eBay achieved record levels of site quality and reliability during a period of accelerating global business growth and new product introductions. Chris is also known for his work in creating and advancing world-class technology platforms at high-growth Internet companies as well as in Fortune 500 environments.

David Mendels
GENERAL MANAGER, MACROMEDIA
Wednesday, February 25, 2004 – 10:00 a.m.

macromedia

SOA+RIA=ROI
Service-oriented architectures are imperative to drive systems and data integration, but for the integration to matter it needs to reach users in meaningful ways. Rich Internet applications (RIAs) combine the richness of the desktop with the reach of the web. Bringing web services and rich Internet applications together fulfills the promise and value proposition of service-oriented architectures, as users benefit from actionable, intuitive, and effective data and information. This next generation enterprise application architecture will deliver solutions that capitalize on existing systems and increase return on investment.

David Mendels is general manager for designer and developer products for Macromedia. Since joining Macromedia in 1992 he's played a major role in the development of Macromedia's product and business strategies and in his current role supervises the gamut of Macromedia's designer and developer products, including Flex, Flash, Dreamweaver, and ColdFusion. In that capacity, he has overall responsibility for the business unit and directly manages product marketing, product management, and product development. Mendels began his career at Macromedia in the international department, moving to Japan to lead the company's field operations, sales, and marketing there. After returning to the United States in 1995, Mendels held positions running business development, alliances and, general management of a number of Macromedia product groups.

Dr. Robert Sutor
DIRECTOR OF WEBSPHERE INFRASTRUCTURE SOFTWARE, IBM
Thursday, February 26, 2004 – 10:00 a.m.

IBM

Web Services: Surviving the Mid-Life Crisis
Mid-life can be tough on those who had fast-paced, high-profile youths but suddenly find themselves with dependents and responsibilities put upon them by others. So it is with Web services as it enters the middle phase of its standardization process and moves from being a favorite of the early adopter crowd to being a dependable citizen for the IT mainstream. How is it adapting to the challenge? Have companies like IBM delivered on the early promises of Web services? Do we need to adjust expectations now for Web services so we're not faced with major disappointments later? Dr. Sutor will discuss these and other issues, as well as explain how Web services is a critical part of how companies can become on-demand e-businesses.

Dr. Bob Sutor is IBM's director of WebSphere Infrastructure Software. He is responsible for business and product leadership for the WebSphere Application Server and WebSphere Studio software offerings. Previously, Bob was IBM's director of Web Services Technology, responsible for driving the cross-IBM Web services initiative to continue and advance IBM's leadership in providing Web services solutions, products, and services.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

DAY 1 — FEBRUARY 24

		Java	.NET	Web Services
8:00		Registration		
9:00		Empowering Java and RSS for Blogging	NET Compact Framework Performance Tips and Tricks	WS-CAF: Standardized Web Services Transactions and Composite Applications
10:00		Opening Keynote – Chris Hjelm, CTO, Orbitz		
11:00		**EXPO OPEN**		
1:00		Keynote Panel Discussion - 21st Century Computing: Is IT Fulfilling its Promise - or Breaking It?		
2:00		Enterprise Architecture & Open Source	Using the Enterprise Instrumentation Framework	Web Services Orchestration, Management, and Security - Will They Play Together?
3:00		Squeezing Java	J2EE/.NET Interoperability	Session TBA
4:00		Opening Night Reception		

DAY 2 — FEBRUARY 25

		Java	.NET	Web Services
8:00		Registration		
9:00		Desktop Java	Best Practices and Techniques for Building Secure ASP.NET Applications	Building Interoperable Web Services
10:00		Keynote Presentation - David Mendels, General Manager of Products, Macromedia		
11:00		**EXPO OPEN**		
1:00		Keynote Panel Discussion - The Open Source Debate		
2:30		Next Phase in Evolution of J2EE	Microsoft Office 2003: A Solutions Platform	Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration
4:00		Aspect Oriented Programming & Java	.NET Framework: Exploring What's New in the Base Class Library for "Whidbey"	Impress the Boss: Roll Your Own Web Services Initiative

DAY 3 — FEBRUARY 26

|		Java	.NET	Web Services
8:00		Registration		
9:00		Simplifying J2EE Applications	Session TBA	Government Real-Time Fraud Detection Using Web Services
10:00		Keynote Presentation - "Web Services: Surviving the Mid-Life Crisis", Dr. Robert Sutor, IBM		
11:00		Apache Axis	Introduction to BizTalk Server 2004 and Web Services	Exploring the Dark Side
12:00		**BREAK**		
1:00		ANT/JUnit	BizTalk Server 2004 Technical Drilldown	ID, Please. The Case for Giving Web Services an Identity
3:00		Session TBA	Moving your Architecture to .NET	Session TBA
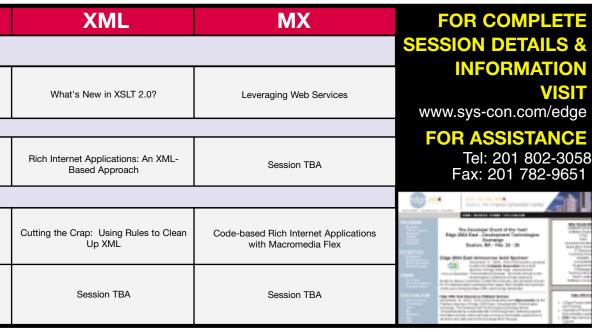
XML	MX
XML Security Integration Challenges	Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex
SOA Foundation Components: Building an XML Content Router	Code Base RIA's
Real Best Practices for XML Web Services Management and Security	Using Flash with Web Services

Microsoft®

Full Day .NET Tutorial Presented by Microsoft
The Smart Client Perspective

XML	MX
XForms - Simplifying the Development of Transactional Web Forms	Using Web Services with ColdFusion
Session TBA	ColdFusion Components
Using XML Schemas Effectively in WSDL Design	Rapidly Build Web Services Applications with ColdFusion and Studio

westbridge

Full Day Security Tutorial Presented by Westbridge Technology
Strategies for Web Services Security Success

XML	MX
What's New in XSLT 2.0?	Leveraging Web Services
Rich Internet Applications: An XML-Based Approach	Session TBA
Cutting the Crap: Using Rules to Clean Up XML	Code-based Rich Internet Applications with Macromedia Flex
Session TBA	Session TBA

FOR COMPLETE SESSION DETAILS & INFORMATION
VISIT
www.sys-con.com/edge

FOR ASSISTANCE
Tel: 201 802-3058
Fax: 201 782-9651

JAVA SESSIONS

Aspect-Oriented Programming & Java
RON BODKIN, NEW ASPECTS OF SOFTWARE

Aspect-Oriented Programming (AOP) has become a hot topic of discussion for enterprise development, with the recent news of support by IBM, JBoss, BEA, Eclipse, and IntelliJ.
• What's real?
• What is AOP?
• What problems can it solve today?
• How can you make an informed decision about using AOP?
• What is the best adoption strategy?
• What are the long-term possibilities for AOP?

This session answers these questions and gives an introduction to AOP for enterprise Java development. AspectJ is a seamless AOP extension to Java. It allows the consistent and flexible implementation of crosscutting concerns such as security, exception handling, testing, logging, caching, and business rules. This is a major improvement on scattered and tangled traditional implementations.

This tutorial introduces AOP and shows how to use AspectJ effectively in an enterprise context, including examples of tools support. Attendees should have experience with OO design and Java development. AspectJ is an open source project available from the Eclipse consortium.

BIO: Ron Bodkin is the founder of New Aspects of Software, which consults and trains on enterprise Java applications with an emphasis on Aspect-Oriented Programming. Ron is also a member of AspectMentor, a consortium of AOP experts. Ron previously worked for the AspectJ group at Xerox PARC, where he led the first AOP implementation projects and training for customers. Prior to that, Ron was founder and CTO at C-bridge, a consultancy that delivered enterprise applications using Java and XML frameworks.

Squeezing Java
ALAN WILLIAMSON, JAVA DEVELOPER'S JOURNAL

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

BIO: Alan Williamson is SYS-CON Media's technical director, and editor-at-large of *Java Developer's Journal*. In addition, he holds the CTO office at n-ary (consulting) Ltd, one of the first companies in the UK to specialize in Java at the server side.

Enterprise Architecture & Open Source
JAMES MCGOVERN, THE HARTFORD FINANCIAL SERVICES GROUP, INC.

Use of open source software within the enterprise is gaining momentum. The vast majority of organizations use some form of open source software, such as Linux, Apache, and JBoss, in production environments. The enterprise architecture, however, needs to incorporate the best thinking of the industry that not only includes using open source but contributing to it. The model in which open source software gets developed has practices that could assist an organization in becoming agile in their software development practices and

allow them to develop software faster, with lower costs and better quality. In this session you will learn:
• Two models of development: the cathedral and the bazaar
• Value proposition of using open source
• Harnessing the power of the mob: the value proposition of contributing to open source
• Making the build versus buy decision: additional thoughts

BIO: James McGovern is an industry thought leader and the author of the best-selling book, *A Practical Guide to Enterprise Architecture* (Prentice Hall). He is working on two upcoming books entitled *Agile Enterprise Architecture* and *Enterprise SOA*. He is employed as an enterprise architect for The Hartford Financial Services Group, Inc. He holds industry certifications from Microsoft, Cisco, and Sun. He is member of the Java Community Process and of the Worldwide Institute of Software Architects.

J2EE v1.4
BILL ROTH, E.PIPHANY

Day-to-day work with deadlines makes it difficult to keep abreast of the rapidly evolving landscape of J2EE, especially given the numerous constituent J2EE technologies. J2EE v1.4 is chockfull of new services that affect and benefit a wide range of enterprise development tasks. This talk will extract core material from the speaker's new *J2EE Developer's Handbook* and describe what's embodied by J2EE v1.4. The new Web services features provided by J2EE v1.4 in particular will be highlighted. The talk will also briefly address those services missing from the current J2EE standards but still needed when building enterprise applications.

BIO: Bill Roth is senior technical evangelist for E.piphany. With over 15 years in this industry, he has played numerous roles in engineering and product marketing including program manager for Java IDL, and managing product marketing and product management for J2EE 1.2 at Sun.

Apache Axis
CHRIS HADDAD, BURTON GROUP

Apache Axis is the very popular SOAP engine, which includes everything you need to start producing Web services. Discover just what Axis is and how you can utilize the power of this free engine to kick-start your Web services.

BIO: Chris Haddad has 13 years experience in Web services, protocols and standards (SOAP, UDDI, WSDL), application framework, development tools, key development languages (Java, C++, Perl, JavaScript), and enterprise application integration. His background includes Grand Central, Flamenco Networks, Adjoin, Securant, Employease, Jamcracker, TRX, and Raygun.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

Empowering Java and RSS for Blogging
JASON BELL

One of the fastest growing areas over the last few years is that of the blogging community. The ease at which you can post and publish information has enabled everyone to become his or her own publisher. One power of blogs has been the syndication of data via the RSS (XML) protocol. Discover how you can easily produce and consume RSS feeds within your Java applications for wider appeal and hook into the likes of JavaBlogs.

BIO: Jason Bell is a Java developer and IT development manager for a B2B portal in the UK. He also contributed to a number of open source projects and is an advocate of everyone reading the API docs.

Ant/JUnit
KYLE GABHART

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

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

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

BIO: Kyle Gabhart is an independent consultant, mentor, and published author, specializing in J2EE, XML, and Web services technologies. He is a prolific writer, with his most recent work displayed on IBM's developerWorks Web site in the J2EE Pathfinder column. You can find him on the Web at www.Gabhart.com.

Desktop Java
JOE WINCHESTER, IBM

Java is enjoying a renaissance on the desktop, and there are several reasons for this:

- The issues that plagued early client/server projects or Java desktop applications have largely been solved. Swing 1.4.2 delivered great performance improvements, good fidelity XP, and GTK look-and-feel classes. Java Web Start now exists as a way to deploy to a client PC the programs that run in a local JRE and enjoy the benefits of local caching, lazy update, and execution within Java's security model
- Java is becoming part of the default installation setup for many PC vendors such as Dell, HP, and Gateway.
- SWT provides a set of rich native controls over and above those offered by AWT, and many Java developers are finding that it's a useful GUI toolkit to use in their end-user applications.

- HTML is no longer being viewed as the only viable client for application development. A number of hybrid technologies that combine the benefits of server-side computing with a client GUI have arisen, including Ultra Lightweight Client, Droplets, JFlash, Macromedia Royale, Asperon, and Thinlets.

This session will look at the Java desktop space and discuss the issues and technologies, as well as what is at stake if Java can or can't recapture its lost pride as a client platform.

BIO: Joe Winchester is a software developer working on WebSphere development tools for IBM in Hursley, UK. Joe was part of the expert group for JSR 57.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

8

WEB SERVICES SESSIONS

WS-CAF: Standardized Web Services Transactions and Composite Applications
ERIC NEWCOMER, IONA TECHNOLOGIES

The Web Services Composite Application Framework is a collection of three specifications – Web Service Context (WS-CTX), Web Service Coordination Framework (WS-CF), and Web Service Transaction Management (WS-TXM) – designed to solve problems that arise when multiple Web services are used in combination ("composite applications") to support information sharing and transaction processing. As coauthor of the specification, Eric will discuss how WS-CAF addresses the underlying issues of Web service context propagation and transaction management to expand the scope, usability, and reliability of Web services for business process automation.

BIO: In the role of chief technology officer at IONA, Eric is responsible for IONA's technology roadmap and the direction of IONA's e-Business Platforms as relates to standards adoption, architecture, and product design.

Government Real-Time Fraud Detection Using Web Services
NEIL MCGOVERN, SYBASE, INC.

Government agencies are faced with increasing amounts of data and are challenged to make sense of, and act on, that data in real time. Failure to interpret and execute on data can result in security threats and, potentially, loss of life. Government agencies are increasingly investing in Web services solutions to address their need for real-time access to information.

The Canadian Passport Office is an example of a government agency leveraging Web services to exchange information in real time to combat terrorism and other illicit uses of fraudulently obtained passports. The agency selected IT consulting firm Pentelar and Sybase, Inc., technologies to electronically authenticate identity document data through the use of Web services and ebXML. Identification tasks performed across 10 provinces and 3 territories previously handled by physical document inspection are now conducted in real time with absolute security and reliability.

In this session, Sybase and Pentelar will discuss this successful pilot project and highlight the ebXML capabilities that enable the Canadian Passport Office to address real-time information exchange.

BIO: Neil McGovern has over 15 years of experience in the software industry. Currently, he is responsible for platform strategy for Sybase's infrastructure product line, including its enterprise-class RDBMS; Sybase ASE; and integration, portal, and tools product lines. Prior to that, Neil was responsible for building the Sybase (New Era of Networks) Adapter suites, commonly recognized as one of the most advanced and complete sets of adapters in the EAI marketplace. Neil developed ERP package expertise as CTO of Convoy Corporation, which specialized in PeopleSoft integration. He led the Development and Professional Services teams for Convoy Corporation, which was purchased by Sybase (New Era of Networks) in 1999 after recording triple-digit growth for three consecutive years.

Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration
DAVID CHAPPELL, SONIC SOFTWARE

Applications are increasingly being developed "built-to-integrate," providing the ability to easily expose key functionality through commonly defined interfaces. Gartner calls this concept SODA, or service-oriented development of applications, fitting into its overall service-oriented architecture landscape. When applied to the ever-present integration challenge, SODA represents a transition to service-oriented integration.

But making the right architectural decisions is absolutely vital to ensuring success with service-oriented integration projects – whether applications were built to integrate or not. Choices at all levels – from application-interface style to overall system architecture – can seriously affect the long-term value derived from integration projects. Application infrastructure products on the market today embody architectural directions that should be carefully examined before making purchase decisions. Any two products that claim to support a "service-oriented architecture" may, in fact, promote radically different architectures.

This presentation will examine the leading choices for supporting service-oriented integration: enterprise service buses (ESBs), integration brokers, and application suite platforms.

BIO: Dave Chappell, VP and chief technology evangelist for Sonic Software, has over 18 years of experience in the software industry. Dave and Sonic Software are involved in many standards organizations, technical committees, and expert groups, including the W3C, OASIS, WS-I, and the Java Community Process. Dave is an active contributor to the Apache Axis project and SOAPBuilders.

Dave also has extensive experience in distributed computing, including message-oriented middleware, CORBA, COM, and Web application server infrastructure. In 2002, Dave received the "Most Outstanding Individual Contributor to the Java Community" award from the editors of JavaPro Magazine.

Securing the Web: What Can Be Done Today
MARK SECRIST, HEWLETT-PACKARD CO.

Security is considered one of the main barriers to the adoption of Web services today. With the proliferation of emerging security standards, there is a lot of confusion over which ones are mature enough to use and how they might fit together. This session will present cover current and emerging security standards for Web services and show how they can be fit together architecturally to address various security concerns. At the conclusion, the audience will have a better understanding of these standards and how to begin securing their own Web services.

BIO: Mark Secrist is a senior software consultant for HP's Developer Resources Organization, with over 15 years of experience working in the software development industry. He provides technical consulting to HP and its customers in the areas of J2EE and Web services development, as well as development strategies for "management-ready" applications. In this capacity, he has written a number of white papers and trade publication articles, presented at a number of industry events, and provided training and consulting to many of our enterprise customers.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

Exploring the Dark Side
PETER VARHOL, COMPUWARE CORPORATION

The growing use of services-oriented architectures puts pressure on application developers relying on Web services for key features of their applications. Performance, scalability, and reliability of these components affect the ability of applications to meet service-level agreements, yet can't easily be analyzed as a part of the application when developers have a problem. In fact, the Web service may be on a different software platform than the rest of the application. This session describes how developers can shed light on memory use in Web services written in either .NET or Java, even if they didn't write the code and wrote their own applications on the other platform.

BIO: Peter Varhol is a product manager at Compuware Corporation, where he provides direction for developer tools targeting enterprise applications and Web services. He holds graduate degrees in computer science and mathematics.

Building Interoperable Web Services Using WS-I Basic Profile
KEVIN LIU, SAP LABS, LLC

The promise of Web services interoperability is based on a set of standards, including XSD, SOAP, WSDL, and UDDI. However, early interoperability experiments such as the SOAPBuilder activity has proven that making these technologies seamlessly work together, inter- or intra-enterprise, is a bigger challenge than most of us expected. Most of the Web services technologies are still in the standardization process, and Web services platforms have to be based on pre-standard draft specifications. The inconsistencies and ambiguities left in these initial drafts can easily lead to different interpretations for the same features and substantially compromise the promised interoperability. This presentation explains how WS-I Basic Profile addresses these challenges, and what are the key advises you can get from the profile that can make your Web services interoperable. It explains why the profile disallows SOAP encoding and why RPC style should give way to document style.

BIO: Kevin Liu is an architect of XML Web services technologies at SAP Labs, California. Kevin helps to drive the adoption of Web services technologies in SAP's strategic products, as well as representing SAP in various standards bodies. He has over 10 years of experience in software engineering and in financial industries and holds multiple master's degrees in information management, finance, and economics.

ID, Please. The Case for Giving Web Services an Identity
ASHISH LARIVEE, NOVELL

Without identity management, Web services can be consumed by anyone. The challenge for Web services developers is to provide appropriate access based on the user's identity. As identity management moves into the forefront of technology, directory services will evolve from simple LDAP repositories used for authentication and storage to robust engines that provide identity integration, access management, and policy enforcement. This presentation will discuss how identity management and directory services provide a robust solution for Web services authentication, authorization, and single sign-on.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms, including Microsoft, Lotus Notes/Domino, and J2EE. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web Application Development Products.

Web Services Orchestration, Management, and Security - Will They Play Together?
PAUL LIPTON, COMPUTER ASSOCIATES

Web Services orchestration, management, and security are among the principal challenges facing implementers of service-oriented architectures today. There is still much confusion in the IT community about the standards themselves, which are at various stages of maturity. Also, their relevance to enterprise IT and how they might someday be able to effectively work together is often unclear. This session provides a useful overview of standards in these three critical areas of Web services – orchestration/choreography, management, and security – and more importantly, how each affects the other. Attendees will gain practical knowledge and a deeper understanding of future trends and the need to address certain real-world issues in order to create a more cost-effective and agile IT infrastructure.

BIO: Paul Lipton is the Web services technology leader for the field services organization and a technology strategist in the Office of the CTO at Computer Associates. He has been an architect and developer of enterprise systems for more than 20 years, and has worked closely with key CA customers to architect distributed solutions using J2EE, .NET, wireless, and Web services technology. Paul has represented CA in various standards organizations, and has participated in the Java Community Process. He has published magazine articles on many technologies including Web services, Java, .NET, EAI, wireless technology, and distributed systems.

Impress the Boss: Roll Your Own Web Services Initiative
BOB ZUREK, ASCENTIAL SOFTWARE

Some companies are leveraging open source software like Linux, JBoss, PHP, MySQL, and Apache SOAP to develop and deploy Web services. As this open source Web services stack continues to mature, companies will benefit beyond simply saving money on license fees; they will be able to assemble the stack and get a Web service up and running quickly as well.

Developers frustrated by their enterprise's lack of Web services interest can immediately master and build Web services at home with open source stack components with little investment except time and a broadband connection. Imagine impressing your company with a demo of wrapping that chunk of code as an open source Web service and calling it from a legacy Visual Basic application.

BIO: Bob Zurek, VP of Advanced Technology, Ascential Software, is responsible for Ascential's overall product strategy. He is instrumental in developing and driving its enterprise integration strategy, including its parallel processing framework, data quality, and Web services strategies. Previously, he was a senior analyst with Forrester Research; prior to Forrester, he held senior management positions at LumaPath, Centive Systems, and Sybase.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

XML SESSIONS

Real Best Practices for XML Web Services Management and Security
KERRY CHAMPION, WESTBRIDGE TECHNOLOGY

Companies deploying Web services in a meaningful way are increasingly finding they need to address Web services management and security early in the architectural phase. Basic Web services connections are easy to do, but managing the security, performance, scalability, and the inevitable changes to the production environment requires some knowledge, expertise, and planning. This session cuts through the hype and outlines real-world mistakes many companies make when deploying Web services and the real best practices from companies who have successfully captured the value of XML Web services. It provides practical advice on how to successfully manage and secure your XML Web service environment.

BIO: Kerry Champion is president and founder of Westbridge Technology, a leading provider of XML Web services management and security solutions. A frequent writer and speaker on XML Web services, Kerry is a renowned technologist who has served in various senior technical and product strategy roles at enterprise security and messaging companies.

SOA Foundation Components: Building an XML Content Router
DAN STIEGLITZ, TIAA-CREF

One of the fundamental components for any burgeoning SOA will be an XML content router. This session explores the concepts, patterns, and open source software available that facilitate building an XML content routing system. The system can be exposed as a Web service or simply as a stand-alone J2EE component for use in your enterprise. The "restaurant" pattern is introduced as the principal design pattern for building the service, and this pattern's applicability to building generic services is discussed. Applying the router as an XML data integration tool is also discussed, as well as its potential for acting as a service orchestrator.

BIO: Visit www.sys-con.com/edge for complete speaker information.

What's New in XSLT 2.0?
STEVE HECKLER, ACCELEBRATE

XSLT 2.0, which may achieve W3C Recommendation status by conference time, offers unparalleled power in conjunction with XPath 2.0 for transforming XML documents. In this engaging, example-rich seminar, Steve Heckler demonstrates the most important new features of XSLT 2.0, including sequences, new data types and XML Schema support, regular expressions, multiple document output, grouping, new control-flow operators, and much more. Current/future support for XSLT 2.0 on the Java and .NET platforms will also be discussed. Most examples will use Saxon, but .NET examples will be included if .NET supports XSLT 2.0 by conference time.

BIO: Steve Heckler has been a leading IT trainer and application developer for nearly a decade. Prior to founding Accelebrate in March 2002, he served nearly seven years as vice president and then president of WestLake Internet Training. He holds bachelor's and master's degrees from Stanford University.

Using XML Schemas Effectively in WSDL Design
CHRISTOPHER PELTZ, HP

Developers building Web services today are beginning to see the value of using the document-style approach over RPC. Recent experience shows that to take full advantage of document style Web services requires a strong knowledge of XML Schemas and related XML standards. This session presents a number of important tips and techniques for properly using XML Schemas in the design of a Web services interface, including important XML-based development tools, binding considerations between XML and underlying objects, WSDL reusability through XML Schemas, and XML Schema naming best practices.

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

Using Rules to Clean Up XML
GARY BRUNELL, PARASOFT

Garbage in, garbage out – it's an axiom that applies to many aspects of enterprise development, but none more so than building reliable and robust Web applications and integration projects with XML. Since its inception, XML has been seen as the cure-all for problems related to Web applications and integration projects. However, poorly written XML can slow down an integration project, or worse, cause the integration project to collapse. The key to successfully using XML in an integration project is to first understand the inefficiencies that may cause poorly written XML, and then apply a rule-based system that establishes policies to follow.

BIO: Gary Brunell joined Parasoft in the fall of 2001 to develop and head the company's professional services division. He is responsible for spearheading delivery services, technical support, and training initiatives as well as establishing process-improvement infrastructure.

XML: Getting Started with Minimum Investment
KETAN PATEL, DATAWATCH

Many organizations are evaluating XML as an enabler for sharing and presenting information. Whether part of a portal strategy, an alternative to EDI for billing, or a data conduit to an information repository, XML is core to many future projects. As IT managers evaluate XML's potential, they should understand its capabilities and limitations. XML is not the panacea for all technology issues. However, as an enabler, XML can provide a common language for sharing information inside and outside your organization. The key is to use the technology in projects that have a clear return and minimize the investments required.

BIO: Datawatch director, Technical Strategy, Ketan Patel has over 15 years of experience developing and marketing technology products to customers who are cautious about technology adoption. During his career, Ketan has focused on bridging the gap between technology and business, and is passionate about bringing information technology solutions to market.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

XForms - Simplifying the Development of Transactional Web Forms

ASHISH LARIVEE, NOVELL

XForms is a W3C specification that specifies a declarative language for solving a common requirement for advanced user interaction, data validation, and XML processing. XForms is designed to be integrated into XHTML, but is not restricted to being a part of that language alone. It can be integrated into any suitable markup language. This session gives an introduction to XForms and explains how XForms in the client tier of the J2EE application architecture. In addition, it will cover the benefits of XForms and why it is a perfect fit for interacting with J2EE and Web services. A demonstration of XForms in a J2EE environment using an XForms- compliant browser and a sample application further illustrates the advantages.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms including Microsoft, Lotus Notes/Domino, and the J2EE platform. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web application development products.

Rich Internet Applications: An XML-Based Approach

CHRISTOPHE COENRAETS, LASZLO SYSTEMS

Rich Internet applications deliver groundbreaking, highly interactive user interfaces that provide a user experience similar and often superior to desktop applications. In this session, we will discuss the benefits of rich Internet applications, review the market landscape, and explore an XML-centric approach for developing rich Internet applications that can be delivered to any browser today. As a practical example, we will demonstrate how to build a rich Internet application version of an Amazon store using an XML-based presentation server to deliver the rich user interface, and the Amazon Web services API to deliver the content.

BIO: Christophe Coenraets worked at Sybase/Powersoft from 1994 to 2000. He started working with Java in 1996 and became the technical evangelist for the company's Java and Internet Application Division. Christophe then joined Macromedia as the company's JRun and J2EE technical evangelist. He joined Laszlo Systems early in 2003.

XML Security Integration Challenges

PHIL STEITZ, AMERICAN EXPRESS

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

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

Attend a FREE One Day Security Tutorial

Presented By

westbridge

Strategies for Web Services Security Success

Wednesday, February 25, 2004

Are you a developer, software architect, IT operator, or security administrator deploying or planning to deploy XML Web services? If so, this technical seminar is designed to arm you with the practical information and best practices you need to securely deploy XML Web services in your environment. Many questions will be addressed, such as: Why do Web services need special security? What standards are being created and what do they address? How do I leverage my existing environment to secure Web services? What are the different architectural and technical approaches to solving the problem? How can I centrally manage security in a decentralized environment?

Course Highlights/Benefits

• Overview of XML Web services security: Why is it important?
• Discussion of various standards (WS-security, SAML, XML-Enc, XML-Sig, XKMS)
• Architectural considerations
• Malicious Web services attacks
• Strategies for securing XML Web services today and in the future
• "Nuts and bolts" demonstrations of security solutions

As part of the tutorial, we will show you how to secure your entire XML environment without adding any additional code.

Course leaders

Dan Neiman is a Senior Systems Engineer at Westbridge Technology and is responsible for training and working with real world customers in solving their security and management needs for XML and Web Services. Previously, Dan worked for Tumbleweed Communications where he directed the field engineers in designing and implementing secure email transports. He has also held positions at CSC and the Department of Defense. Dan holds a BS degree in Management and Decision Systems from Carnegie Mellon University.

Peter Bostrom is a Federal Technology Director at Westbridge Technology and works closely with federal customers. Previously, Peter was VP of Product Management at Sequation, a security solutions provider and CTO and co-founder of ExactOne, an Internet-scale distributed search engine. Peter also worked in a variety of sales, consulting and product management roles at Legent Corporation. He was awarded Systems Engineer of the Year award in his first year there. Peter can be heard on National Public Radioís WAMU in Washington, DC as a regular guest on íPublic Interestî hosted by Kojo Nnamdi and is president of the DC Association of Internet Professionals. Peter holds a BA in International Affairs from The American University after spending three years as an Army Ranger.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

MX SESSIONS

Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex
KEVIN HOYT, MACROMEDIA

Learn how Macromedia's technology initiative, Flex fits seamlessly into today's new service-oriented architectures (SOA). We'll cover design patterns for rich clients, accessing web services and securing your Flex application.

BIO: As a senior sales engineer for Macromedia, Kevin Hoyt plays a primary role in evangelizing the product line, educating the customer base, and supporting the sales organization. Kevin is also a regular speaker at various user group forums throughout the United States.

Leveraging Web Services
TOM JORDHAL, MACROMEDIA

Web Services technology is changing the way we think about designing and building applications. Come and learn what all the fuss is about, find out exactly which problems web services solve, see web services created and used, and even discover how web services expose the world of .NET.

BIO: Tom has been a Principal Engineer at Allaire/Macromedia for over 6 years and has spent the last 2 years on the front lines of Web Services. He is one of the implementers of Apache Axis and is the Macromedia representative on the W3C Web Service Description WSDL 1.2 working group. Before getting involved in Web Services, he was the technical lead for the ColdFusion on UNIX products. Tom is currently 'back home' working on ColdFusion.

ColdFusion Components
GLENDA VIGOREAUX, BUSINESS SYSTEMS CONSULTANTS, INC.

ColdFusion Components combine the power of objects with the simplicity of CFML. This is the way object-based development was intended to be, and in this session, you'll learn about this combination first hand. Starting with a simple data-driven application, you'll gradually convert it into a highly scalable and manageable multitier application, and in the process, will amaze yourself at just how easy ColdFusion makes this process.

BIO: Glenda Vigoreaux has been a Certified Macromedia Instructor and Developer for the last 5 years. She is currently heading the web development division for Business Systems Consultants, Inc. the most successful reseller and services provider of software for associations in the greater Chicago area.

Rapidly Build Web Services Applications with ColdFusion and Studio
SIMON HORWITH, ETRILOGY

The last year has shown that web services are not just another passing fad and their promise of platform-independent distributed applications has been realized. Compared to other application server platforms, ColdFusion makes creating web services easy. This session covers how to create a ColdFusion Component (CFC) in Dreamweaver, as well as how to expose that CFC as a web service by just toggling one attribute of the CFC. That's right: in ColdFusion, it is just that easy.

BIO: Simon Horwith is Chief Technology Officer of eTRILOGY Ltd., a software development company based in London, England. Simon has been using ColdFusion since version 1.5 and is a member of Team Macromedia. He is a Macromedia Certified Advanced ColdFusion and Flash Developer and is a Macromedia Certified Instructor. In addition to administering the CFDJ-List mail list and presenting at CFUGs and conferences around the world, Simon has also been a contributing author of several books and technical papers.

Code-based Rich Internet Applications with Macromedia Flex
CHRISTOPHE COENRAETS, MACROMEDIA

Learn how to use Macromedia's technology initiative Flex to create rich internet applications. This session will cover using components, layouts, and managers to build user interfaces as well as using Flexs XML-based language to create and manipulate client-side data models.

BIO: Please visit www.sys-con.com/edge for a complete speaker bio.

Using Macromedia Flash with Web Services

Web services, a technology that allows developers to execute remote procedures, are emerging as a revolutionary tool for web application development. Macromedia Flash MX 2004 Professional is a powerful tool for building applications that consume web services built in any technology, including Macromedia ColdFusion, Java, ASP.NET and PHP. In this session you will explore the visionary computing model that web services represent as you use Macromedia Flash components to develop a web service based application. You will learn how to discover web services, work with data and UI components, perform data binding, examine security issues and aggregate multiple web services into a cutting edge web service consumer.

For a Complete List of MX Sessions please visit www.sys-con.com/edge

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

For Assistance Tel: 201 802-3058 Fax: 201 782-9651

CONFERENCE: Feb. 24 – 26, 2004   EXPO: Feb. 24 – 25, 2004

Hynes Convention Center, Boston, MA

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

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

Please complete sections 1, 2, 3 and 4

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

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

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

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

Check # _____ Amount of Check $ _____

Charge my   ☐ Visa   ☐ MasterCard   ☐ American Express   ☐ Discover

Name on card _____

Card # _____ Exp. Date _____

Signature _____

Billing Address (if different from mailing address)
_____

3 PLEASE INDICATE YOUR CONFERENCE CHOICE

Total Registration fee $_____

Conference Choice	Before 12/19/04	Before 1/24/04	Before 2/21/04	Onsite
☐ **GP: Gold Passport Three-Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions & one day of J2EE training.	$1,395.00	$1,495.00	$1,695.00	$1,795.00
☐ **3D: Three Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$1,295.00	$1,395.00	$1,495.00	$1,695.00
☐ **2D: Two Day Conference** Good for two days of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$995.00	$1,195.00	$1,295.00	$1,395.00
☐ **1D: One Day Conference** Good for one day of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$595.00	$695.00	$695.00	$795.00
☐ **TRAINING PASS:** Good for Developing J2EE Applications Java Training Session only. Select one: ☐ February 25th ☐ February 26th	$495.00	$595.00	$595.00	$695.00
☐ **VIP PASS FREE with preregistration** Select one: ☐ FREE - The Smart Client Perspective (Feb. 24) ☐ FREE - Strategies for Web Services Security Success (Feb. 25)	FREE	FREE	FREE	$99.00
☐ **EO: Expo Hall Only** Includes keynotes, expo presentations, and workshops.	FREE	FREE	FREE	$50.00

SPECIAL DISCOUNTS AVAILABLE
Take advantage of the Early Bird and Preregistration values available right now, or save even more with a group of 5 or more. For special group discounts contact Michael Lynch at mike@sys-con.com, or by phone at (201) 802-3058.

4

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

B. Business/Industry

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

C. Total number of employees at your location and entire organization (check all that apply):

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

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

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

E. What is your company's gross annual revenue?

☐ $10 billion or more
☐ $1 billion – $9.9 billion
☐ $100 million – $999 million
☐ $10 million – $99.9 million
☐ $1 million – $9.9 million
☐ Less than $1 million
☐ Don't know

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

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

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3058 by February 10, 2004.

SYS-CON EVENTS

CANCELLATIONS, SUBSTITUTIONS, REFUNDS
Fax written request to SYS-CON Registration 201-782-9651. Requests for refunds received prior to January 16, 2004 will be honored less a 10% handling charge; requests received after January 16, 2004 and before February 6, 2004, will be honored less a 20% handling charge. No requests for refunds will be honored after February 6, 2004. Requests for substitutions must be made in writing prior to February 20, 2004. No one under 18 is permitted to attend. No warranties are made regarding the content of sessions or materials.

Speakers, sessions, and schedule are subject to change without prior notice.

No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

SPONSORS & FEATURED EXHIBITORS

Computer Associates
Gold Sponsor

Computer Associates International, Inc. (CA) offers the most comprehensive portfolio of solutions for managing Web services and Web services platforms. CA's end-to-end solutions, including Unicenter® Web Services Distributed Management, dramatically reduce the cost and complexity of Web services deployments by providing visibility and control, optimized performance and increased enterprise responsiveness. For more information, please visit ca.com/webservices.

EAI Industry Consortium
Association Sponsor

The EAI Industry Consortium, a non-profit 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 a venue to develop, create and debate.

Ektron

Ektron is a leader in developer-friendly Web content authoring, publishing and management for non-technical end users. Ektron's easy-to-deploy, affordable software enables users to easily update, publish and manage Web content from anywhere without knowing HTML or XML, while working in an environment easily controlled by Web professionals. Nearly one million individuals worldwide use Ektron's award-winning products.

IONA

Artix is a family of Web services based service-oriented integration products that renovate IT assets and consolidate legacy middleware to reduce the complexity and cost of IT operations. Customers use Artix to maximize the value of existing systems through the deployment of best of standards-based middleware and enterprise Web services.

Kenetiks
Education Sponsor

Kenetiks is a worldwide provider of Java-WebSphere training and project mentoring with courses available for every skill level. Our Kenetiks Academy can show you how to build your own J2SE/J2EE applications for IBM WebSphere Application Server and IBM WebSphere Portal Server using WebSphere Studio Application Developer.

Macromedia
Platinum Sponsor

Macromedia is motivated by the belief that great experiences build great business. Macromedia provides the industry-leading software that empowers developers and designers to create and deliver those effective, compelling, and memorable experiences.

Microsoft
Gold Sponsor

Microsoft .NET is a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of Web services-small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.

Novell
Silver Sponsor

Novell helps customers dynamically access, provision, manage and deliver secure information services. Customers can flexibly integrate applications and data and ensure unified identity across systems. Novell's Web Services platform, exteNd, is a comprehensive, integrated approach to the visual development and deployment of services-oriented applications.

OASIS
Association Sponsor

OASIS is the non-profit, 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.

Sonic Software

Sonic Software is changing the economics of integration. Sonic provides the first comprehensive business integration suite built on an enterprise service bus (ESB). The Sonic product line delivers a distributed, standards-based, cost-effective, easily managed infrastructure that reliably integrates applications and orchestrates business processes across the extended enterprise

Velocitis

Velocitis provides the next level in agile software development tools with Flywheel, a code centric tool for rapidly designing, visualizing and refactoring Visual Studio .NET code. Fine grained synchronization allows the Flywheel and Visual Studio tools to be used in any order, at any time in the development process. With Flywheel developing in code or visually is the same because, the code is the model.

Westbridge Technology
Education Sponsor

Westbridge Technology provides Web Services Management solutions that enable enterprises to secure and manage XML Web Services networks. The Westbridge XML Message Server (XMS), enables organizations like the FTC, Danish Government and Travelers Property Casualty to seamlessly connect, secure and manage XML Web Services networks. For more information, visit www.westbridgetech.com.

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Java and Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. SYS-CON Publications, Inc., is independent of Sun Microsystems, Inc. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.

XMLSPY2004 and
MAPFORCE2004
by Altova, Inc.

Reviewed by
Vijay Phagura

I f you are an enterprise or XML developer and wish you had a tool that could make your life easier, check out this review as it discusses two of the most useful tools in the industry. XMLSPY is more than an editor; it's a toolbox for XML. MAPFORCE is another XML application that provides different facilities to a developer or a designer.

XMLSPY2004

This is an XML workbench that includes a lot of features. I'll describe some of these features here; for more information check out Altova's Web site.

Document Creation

I like document creation because you don't have to create a project to create a document or a schema, though you can add the created document or schema to a project later. Note that in the New menu item of XMLSPY there are many types of documents that you can create with it.

After a document type is created the tool gives the user different views of it, which is helpful and easy to use. In the case of an XSD, it's viewable in the form of a block diagram and the user can navigate the blocks to see the properties of each block in different property windows. Parent and child relationships can be collapsed or expanded; blocks can be easily made reusable without typing in the Text view.

Documentation

This is one of the few good things in life! After you create a schema, click the Generate Documentation menu item and the tool will create the documentation for your schema. You have two choices for the document format: Word or HTML. They both worked well for me.

The documentation generated is pretty good with diagrams and links to navigate to the various components. After you edit the schema, you have to regenerate the documentation every time as it doesn't occur automatically. If the schema is reasonably large, as they often tend to be, this feature may turn out to be time-consuming.

XSD to the Database and Back

This is one of the useful features in XMLSPY and can turn out to be very productive in the design phase as well as the maintenance phase of a project. If you design an XSD, it can be converted to a database schema. Of course, the database types are limited, but it covers most of the popular databases like Oracle, MS SQL Server, etc. You may choose to design a database using this tool by creating an XSD and then creating the database schema.

An XSD can also be created from an existing database; as noted earlier, it is limited on the databases but it still covers most of the popular ones.

MAPFORCE2004

This is a good mapping tool for XML. MAPFORCE2004 can be considered an additional tool that reinforces XMLSPY. It's easy to use and installs fairly simply.

Mapping at Its Best

You can easily map XML to XML or the database to XML. The mapping is very pictorial; the XSD or the XML is shown as a block with all its elements and attributes.

The process of mapping is made very visual and is just connecting lines from the source to the target blocks. As soon as
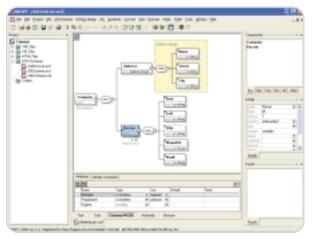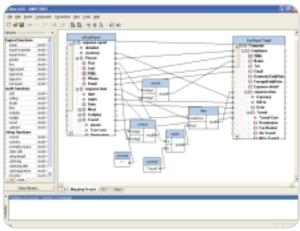

Figure 1 XMLSPY2004

Figure 2 MAPFORCE2004

Vijay Phagura, a professional Java/J2EE consultant, has over 12 years of experience in software architecture and development. He specializes in designing and developing software using J2EE and other Java technologies.

vphagura@yahoo.com

you connect or map elements or attributes, an XML text is being written for you. There is a Text tab that shows the generated XML.

This tool also gives the user many functions for mapping. These functions are categorized as Logical, Math, Strings, etc. For instance, the user can use the concat String Function, so that while mapping an element or attribute an additional string can be concatenated to the mapped tag. Apart from the functions, there are also filters that can be used while mapping.

It supports database mapping too. The following databases are currently supported:
- Microsoft Access 2000
- Microsoft SQL Server
- Oracle

This mapping also follows a similar mapping approach, as described above.

Transformation and Code Generation

This tool provides options to generate XSLT from the resulting XML file. This is very helpful and makes it easy to reuse the generated XML in other components of a project. You can also use XMLSPY with the XSLT to do other things with it.

There is also an option to generate Java code, which can be compiled and used along with your project. It makes a JBuilder project file and an Ant build script, which makes building the generated code easy. However, you may still have to do some work to include this build in your project's overall build script.

Summary

Overall I found both these tools to be very useful, and they can save you a lot of development effort. If applications involve a considerable amount of XML and transformations, then I certainly suggest using them.

The conversion of XSD/XML to the database and vice versa can be done using both of these applications, but XMLSPY does this at development time or statically, while MAPFORCE does it dynamically or at runtime, of course, after the mapping has been done once!

Altova, Inc.

900 Cummings Center
Suite 314T
Beverly, MA 01915-6181
Phone: 978 816-1600
Web: www.altova.com

Specifications

Platforms: Windows XP/2000/NT/98/ME

Test Platform

Win XP Pro, 2GHz CPU, 512MB RAM

Snapshot

Target Audience: Software developers and architects
Level: Beginner to advanced
Pros:
- Powerful editor
- Easy to use
- Configurable
- Powerful conversion
- Powerful mapping
- Powerful code generation

Con:
- Needs more integration between the two tools

From Within the
Java Community Process Program

Onno Kluyt

Where we've been and where we're going

Welcome to the January edition of the JCP column. In the time-honored New Year's tradition of offering a perspective of the past year, I have written a different column on the JCP than in other months. Driven partly by the fifth anniversary milestone and partly by some interesting trends happening around us, I'm taking this opportunity to chat with you about this past year and about what may lie ahead for the Java community in 2004.

A Look Back

To start with the technology side of things: 2003 saw the J2EE 1.4 JSRs go final. The Executive Committee voted unanimously in favor, thus giving a strong endorsement to this new release of the J2EE technology. This version took about six months longer to complete than planned due to the community's request to wait for the completion of the WS-I Basic Profile and include it in the J2EE 1.4 technology release.

The milestone event for the J2ME technology was that JSR 185 (Java Technology for the Wireless Industry) went final in July. This year also saw the submittal of new JSRs that will introduce various new features to the Java programming language and new JSRs that aim to ease the development of J2EE-based applications.

While the year is not yet over as I'm writing this, 35 new JSRs were submitted during 2003. This brings the annual average of new JSRs to 47 per year. Of the total 237 submitted JSRs, Sun is leading with 44% with the majority of JSRs being led by other members of the community. With the completion of J2EE 1.4, 84 JSRs have finished, meaning that the JCP members completed over a third of all the initiatives they started. Out of 237, 24 JSRs were withdrawn by their submitters or rejected by the Executive Committee.

Late October 2002, JCP 2.5 was launched, which was accompanied by a new membership agreement, aka JSPA 2. This new agreement gave open source an official standing within the JCP and required that TCKs be available free of charge to nonprofits and qualified individuals (among many other things). Since then the individual membership within the JCP has been on the rise (although the abolishment of the fee may have had something to do with it), so much so that now there is an equal number of corporate and individual members.

Onno Kluyt is the director of the JCP Program Management Office, Sun Microsystems.

onno@jcp.org

Also noteworthy are Richard Monson-Haefel's election to the Executive Committee, so now there are two individuals serving on that body; JBoss Group's joining of the JCP as well as MySQL and even Alan Williamson. The ApacheCon attendees singing "Happy Birthday" was memorable as well!

A Look Ahead

Early in the new year you can look forward to the launch of JCP 2.6, which focuses on making expert groups more accessible and the overall process more transparent, and achieving a higher uniform quality of TCKs. The JCP.org will be revamped in the area of support, providing among other things a place for community members to find early drafts, notes, and observer aliases for the JSRs operating under this new version.

Let's take a closer look at the individual membership for a moment. By JavaOne 2004 it's likely there will be more individuals who have signed the JSPA than companies. One potential consequence could already be observed during this year's EC elections: eight of the nine candidates for the SE/EE EC were individuals. Most spec leads I talk to highly value having individual members on their expert groups. These developers are often very motivated to have the results of the JSR because they are committing personal time. However, while this provides opportunity it can also bring risk with it. Other organizations have failed in the past over not properly balancing individual interests versus corporate interests.

A current discussion among EC members is the flexibility that the process provides to spec leads in choosing a licensing model for the output of a JSR. Historically, this flexibility was built in to encourage JCP members to take up the role (and the burden) of spec lead so that many shoulders would carry the load, if you will. This appeared to have succeeded in that about 55 different JCP members lead one or more JSRs. However, this flexibility can lead to some complexity on the side of the implementer where multiple JSRs are often combined into one product. Can a different balance be found between flexibility for the spec lead versus flexibility for the implementer that still maintains the spec lead role as an affordable one while also ensuring ample choice for the implementer in selecting a business model for its products?

Speaking of business models and licenses, I want to bring the Creative Commons (www.creativecommons.org) to your attention. This initiative was started by Lawrence Lessig and others. The principle on which it is based is very similar to our key principle with the Java technology. Creative Commons attempts to explore the space between the typical copyright statement of "all rights reserved" and the public domain that says "no rights reserved." In other words it is about "some rights reserved." This is very similar to how the JCP manages the Java technology; you get to benefit from the efforts of the community provided that you promise to do so in a compatible manner. Visit the Web site and explore this notion for yourself. It is one of the things I'll be doing in 2004.

That's it for this month. I'm very interested in your feedback. Please e-mail me with your comments, questions, and suggestions.

Next **Month**

Point/Counterpoint: The Art of the UI

Join the debate as Alan Williamson and Joe Winchester present their views on where the Java UI's future lies: thin clients (i.e., browsers using a "simple" HTML interface) versus thicker clients (Flash, applets, WebStart, etc.).

Java Technology on Mac OS X

You can't think about Java on the desktop without giving serious consideration to the Mac OS X implementation. For many consumers, Apple has defined the desktop experience over the past twenty-some years. Many of the expectations you have of desktop applications were shaped by the work at Xerox PARC, Apple, and later Microsoft. Each of the more than seven million copies of Mac OS X runs J2SE 1.3 or better – a potential market for your desktop application that is configured and ready to run it.

The Delegation-Managed Persistence Entity EJB: A Composite Entity Bean for a New Generation

With the introduction of the EJB 2.0 specification, the classic composite entity bean design pattern became outdated overnight. This article presents a new pattern that can serve as a proper replacement. This pattern is called the delegation-managed persistence bean and allows developers to represent objects that span multiple database tables. DMP beans provide a better solution than the original Composite EJB pattern, without making you roll your own persistence mechanism.
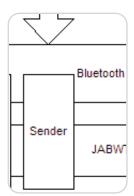
Go Wild Wirelessly with Bluetooth and Java

As a frequent visitor to J2ME and Bluetooth developer forums, I've noticed that one of the most frequently asked questions is "How do I get started with Bluetooth?" Despite its introduction in 1998 and a highly hyped year in 2001, Bluetooth application development remains hideous and challenging for lots of developers. This article will take you to the next step of Bluetooth development and look at a newly developed specification that can assist Bluetooth developers in building applications rapidly.
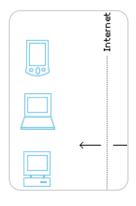
Product Review
Droplets by Droplets, Inc.

Tired of using that Web access e-mail client? Going crazy filling out those expense reports using a clunky Web-based system? Missing the superior and responsive desktop application? You're not alone! According to usability research firm Nielsen Norman Group, "Billions of dollars are wasted yearly in lost productivity as people wait for Web pages to perform duties that could be better handled by a 1984 Macintosh-style GUI application." The Droplets application platform, with User Interface Server and SDK, enables developers to create software that combines the convenient, instant deployment benefits of the Web with the high usability and rich UI of traditional clients.

LINUXWORLD
CONFERENCE & EXPO ®

Where
Open Minds
Meet

Chris Stone
*Vice Chairman, Office of the CEO,
Novell, Inc.*
Wednesday, January 21
9:15 am – 10:00 am

Dave Dargo
*Vice President, Linux Program Office,
Oracle*
Wednesday, January 21
11:45 am – 12:30 pm

Tom Killalea
*Vice President of Infrastructure,
Amazon.com*
Wednesday, January 21
2:45 pm – 3:30 pm

Sam Greenblatt
*Senior Vice President and Chief Architect,
Computer Associates Intl.*
Thursday, January 22
11:30 am – 12:15 am

Ross A. Mauri
*General Manager, e-Business on demand,
IBM Systems Group*
Thursday, January 22
1:45 pm – 2:30 pm

CONFERENCE: January 20 – 23, 2004

EXPO: January 21 – 23, 2004

THE JAVITS CENTER: New York, NY

LinuxWorld Conference & Expo is the world's leading and most comprehensive event focusing on Linux and open source solutions. This January, discover how companies across the globe have achieved higher profits and increased their productivity by utilizing Linux, the fastest growing operating system in the world.

- Get in-depth coverage on the latest Linux and open source developments and learn about innovative product solutions from the industry's top companies.

- Network with the leaders in the open source movement and discuss with your peers how to best leverage the technology for your organization.

- Participate in LinuxWorld's world-class education program and benefit from interactive training in the all-new Hands-on Labs!

- Attend the 2nd annual Linux Financial Summit and hear how Wall Street firms are leveraging Linux to achieve a lower total cost of ownership. (Sponsored by BEA)

- Discover real world practical solutions and find answers to today's most critical IT issues.

REGISTER ONLINE WITH REFERENCE CODE: A-JDD

IDG WORLD EXPO

Plenty of companies are already following this path, and will continue to invest in scaling out their IT networks by leveraging the economics of industry-standard x86 computing, where Solaris x86, Linux, and open source alternatives offer up a pretty compelling story. At the same time, there'll be a renewed focus on scaling up, leveraging highly available 64-bit Unix systems for demanding government, enterprise, and service provider environments, as companies dial-up IT investments to keep ahead of the competition.

4. Software licensing and deployment models will be radically simplified. 2003 was the year we saw a bunch of companies finally get the service provider model right. Companies like Salesforce.com, eBay, and Google are in the software business, but they don't sell their software, they let you use it or rent it. You're going to see a lot more activity in this space in 2004.

You're also going to see a big upset to the dominant software players in the marketplace, with game-changing innovations like the Sun Java Enterprise System and Sun Java Desktop System. The Java Enterprise System is just $100 per employee per year. China is adopting the Java Desktop System for the entire country – in a deal that could eventually reach a half-billion desktops. That's big news.

5. The typical data center out there is like a fingerprint – no two are the same. They're all built out of hundreds of hardware and software parts and integrated onsite by an army of high-priced consultants. This model is going to fall apart as enterprise IT folks are under increasing pressure to deploy new applications and services quickly and cheaply. They're going to look instead to purchase complete systems, proven designs, tested infrastructure, and higher levels of integration through R&D, all done on the IT vendor's nickel instead of the customer's. ✐

McNealy announcing the news that China is adopting the Java Desktop System for the entire country – in a deal that could eventually reach a half-billion desktops.

PHOTO COPYRIGHT 2003 SYS-CON MEDIA

Where Is *i*-Technology
Going in 2004?

"So where is it all going?" It's the question every technology sage is always asked, and yet, of course, it's the question that's the most difficult to answer.

Compiled by **Jeremy Geelan**

Here at *JDJ* we decided nonetheless to ask a welter of the brightest and most prescient *i*-technology professionals what they are seeing, what they think the individual pieces in 2004 are, and how those pieces fit into the bigger picture.

Java features prominently, Web services (and associated technologies) looms large, and SOAs (service-oriented architectures) make their debut, building on the Web services buzz but taking us to the kind of loosely coupled, asynchronous, message-oriented paradigms much favored by enterprise architects.

In these globalized times, as the three E's – Excitement, Expertise, and Energy – return to the technology market, it is apparent from the comments that follow that many of the innovations of the last few years are expected at last to bear fruit in 2004, and that the industry's stakeholders – software vendors, developers, technology investors, and users alike – are looking forward to the year ahead with renewed vigor.

What are the common themes that our selected seers identified? We'll let the respondents speak for themselves.

We hope you enjoy what they have to say.

Internet 2.0

Jeremy Allaire, cofounder with his brother "J.J." of Allaire Corporation, which later merged with Macromedia – and now Technologist in Residence at General Catalyst Partners, Boston – says that "We're in the midst of a new convergence of trends that are resulting in what I call Internet 2.0."

Allaire continues: "Like the trends that converged to form the first-generation Internet (commodity PCs, ubiquitous and affordable connectivity to IP networks, prevalence of LANs in corporations, widely available design tools, the introduction of HTTP/HTML and the Web browser, etc.), these trends are mutually reinforcing, creating an accelerating dynamic."

Allaire's top 10 trends, those he says he's "watching closely and that appear to continue gaining ground," are in Table A.

'Repurposing' Applications, Not Inventing Them

Though not a venture capitalist like Allaire, David Litwack – as president and chief executive officer of SilverStream Software, now a senior vice president with Novell and a member of its Worldwide Management Committee – has a wealth of experience in Internet technologies of every stripe. With complete responsibility for Novell's Web Application Development Products, Litwack is well positioned to survey the present and future *i*-technology scene.

"Too much is made of the specific standards associated with Web services," Litwack observes. "While these are important, it misses the forest for the trees. The key trend is that we are building fewer and fewer pure new applications and instead are increasingly repurposing what we have."

The mechanism for doing this, Litwack explains, is "to virtualize, to create composite or logical applications in front of our physical systems or, in short, to create services."

These services, he says, may sometimes be SOAP/WSDL based and generally will be XML. "But the principle remains the same. Create a logical data source that is constructed from one or more physical systems that represent a real business function as perceived by some audience. The effect is to present to the user an application that looks like it was developed just for them – everything they need to do their job and no more than they need."

That, according to Litwack, is the future. His other, more specific predictions can be read in Table B.

Outsourcing in the U.S. Will Fade

From open source to outsourcing, Adam Kolawa, the charismatic and forward-looking president of Parasoft, believes that no theme will be more important in 2004 than "BPO" – business process outsourcing.

"In about a year," predicts Kolawa, "companies will start to see that outsourcing is not working for them as they had hoped.

"Yes," he says, "there's a scare within the industry right now and developers are afraid of losing their jobs, but companies will see that the quality of code developed offshore isn't getting any better and they will end up spending just as much to fix the code as they would have if it was developed by their own team."

There is, nonetheless, another side to the story: "I have had several encounters with other U.S. companies," Kolawa observes, "and I think the productivity of developers is staggeringly low."

> "This model is real, deep, and transformative – most of the interesting businesses that I'm involved with are centered on managed services, not licensed software"
>
> —*Jeremy Allaire* on *Hosted Apps*
> Technologist in Residence, General Catalyst Partners, Boston

Jeremy Allaire: Rich client apps, Web services, broadband, real-time

1. *Broadband adoption:* We're crossing the 50% mark of the online user threshold.
2. *Wireless adoption:* In global (e.g., cellular), local (e.g., Wi-Fi), and personal (e.g., ZigBee, Bluetooth), we're seeing a resurgence in wireless capabilities.
3. *Rich clients:* Rich Internet applications built around client runtimes such as Macromedia Flash Player are transforming the application and content landscape.
4. *Web services:* This architecture is now pervasive and we're finally seeing business models built around it.
5. *Real-time communications:* VoIP, rich media real-time collaboration, and the integration of IM and SMS into our application infrastructures.
6. *Devices:* The proliferation of media-enabled, Wi-Fi connected devices is growing by leaps and bounds.
7. *Big data:* Storage and bandwidth price/performance continues to outpace all expectations, driving new application capabilities/demands.
8. *Hosted apps:* This model is real, deep, and transformative – most of the interesting businesses that I'm involved with are centered on managed services, not licensed software.
9. *Paid content:* Subscriptions and micro-payments are now emerging as real economic models for the Internet and content.
10. *Software manufacturing economy:* Global outsourcing, open source, Web services, and ASP delivery are wreaking havoc on the old world of "industrial software," and reducing costs and changing the high-tech labor landscape.

Jeremy Allaire
Technologist in Residence
General Catalyst Partners, Boston   Table A

David Litwack: Identity management, the open source movement

1. To have services that are delivered to the individual in a secure and relevant way, we have to know quite a bit about identity. But electronic identity is generally fragmented across the enterprise.
2. Identity is the yin to the Web services yang. In effect, we are moving the metaphor from client/server, the physical device talking to a specific application, to identity/services – the person, regardless of how he or she connects, to the business function, regardless of how it's implemented physically. In the purest sense, this is technology finally adapting to the individual, rather than the other way around.
3. The open source movement may not be directly related to the above items, but it will be a powerful force in moving the industry toward them. Historically, new technologies are best adopted during a platform shift. The economics of open source and Linux are compelling.
4. As a result there will be a large scale movement in this direction. As applications are built or migrated, they will be more likely to adopt newer technologies. The best applications to move will be those based on cross-platform Java/J2EE.
5. .NET applications will move more slowly, due to a higher loyalty to Microsoft, but open source projects like Mono will enable .NET applications to run on Linux.
6. Few Web services will be open source, since these frequently are tied to strategic, proprietary systems. But the presentation that consumes the Web service will be open sourced, especially via portlet standards like JSR 168 and WSRP. The back-end services remain insulated, but users will have lots of flexibility in how they use them.

David Litwack
Senior Vice President
Web Application Development Products, Novell   Table B

He has seen statistics from which it appears that, in the U.S., developers are responsible for about 5,000 –10,000 lines of code each – "which is about a factor of 10 too low to really be productive," he says.

"Something has to change," Kolawa adds, "because if they continue to produce so little, these developers will be out of jobs. The rate of pay developers receive doesn't justify the amount of code they produce."

But BPO, Kolawa insists, will turn out not to be the answer. He offers an example to back up this prediction. "One architect I met recently had a team of 32 programmers working on 100,000 lines of code. They worked with a company in India who claimed to be CMM Level 5 certified. However, when they got their code back it crashed. Apparently the company in India didn't understand the specs."

'Client/Service' Is What's Next

Mitchell Kertzman makes some very interesting prognostications, as befits someone with an almost unique background.

The former CEO of Sybase and co-creator of PowerBuilder is now, in true gamekeeper-turned-poacher fashion, a venture capitalist. From his position as a partner with Hummer Winblad Venture Partners, in San Francisco, he looked toward the new year for us.

"My new view on the industry from the venture capital seat is absolutely fascinating," says Kertzman. "The rumbling about the end of innovation in technology is certainly not true, particularly in software."

Kertzman's thoughts on 2004 and beyond are in Table C. "There are a couple of broad technologies," he says, "that I think will see lots of excitement in 2004."

> "I think that Web services and other service-oriented architectures may finally fulfill the promise of distributed application architectures that we've all been pursuing for years"
>
> —*Mitchell Kertzman* on *Client/Service*
> Partner, Hummer Winblad Venture Partners

see continued belt-tightening," he adds.

"We will see more discretionary spending on *i*-technology products and services," Gupta continues, "but we will also see relentless outsourcing.

"In other words, we'll see further belt-tightening in some areas such as proprietary Unix servers and possibly even the needless upgrade cycles of some software such as MS Office, but we will see a willingness to spend more on software to provide visibility and control in operating an enterprise."

> "Clearly the most culturally important thing going on at the moment is the explosion of Weblogging and syndication technology – those two trends are related but certainly separable"
>
> —*Tim Bray* on *Blogging*
> CTO/Founder, Antartica Systems

Relentless Outsourcing

Like Kertzman, Rajiv Gupta is one of the leading lights of the technology world. Now CTO of Confluent Software, Gupta is the serial innovator once responsible for HP's "E-Speak" – back in 1999, which was years before its time, making him a perfect choice when it comes to looking forward.

As far as Gupta is concerned, he believes that the technology slump is over for now. "But that does not mean that we will not

The remainder of Gupta's observations are in Table D.

Weblogging and RSS

Tim Bray, as co-inventor of XML, is someone who knows how to set his watch five minutes ahead of other people's.

Not surprisingly, then, XML-based RSS feeds appeal to him, to the extent that he says, without hesitation: "Clearly the most culturally important thing going on at the moment is the explosion of Weblogging and syndication technology."

The two trends are "related but certainly separable," says Bray. "The commercial impact of all this is just impossible to predict," he continues, "but the world in 2005 won't look at all like it did in 2001, and syndication [during 2004] is going to be a big part of the difference."

Some of the other predictions Bray has for 2004 and beyond can be found in Table E.

Massive Marketing Spending by MS

Javalobby founder Rick Ross was a natural expert to turn to for a forward-looking exercise like this. Ross can always be relied upon to keep focus, and in Java terms that means keeping an eye on "the competition" – and sure enough he observes "Microsoft will continue to spend more on marketing than most of their 'competitors' generate in total revenue."

> "To have services that are delivered to the individual in a secure and relevant way, we have to know quite a bit about identity. Identity is the yin to the Web services yang"
>
> —*David Litwack* on *Identity Management*
> Senior Vice President, Web Application Development Products, Novell

Mitchell Kertzman: RSS, Web services, client/service

1. I've been following RSS, the publish-and-subscribe technology based on XML that is the foundation of the syndication of Weblogs.

2. I don't think that the Weblogs themselves are the big opportunity; I think RSS can have a role in application architectures as well.

3. As an old PowerBuilder guy, involved in the early days of client/server, I think that Web services and other service-oriented architectures may finally fulfill the promise of distributed application architectures that we've all been pursuing for years.

4. My friend Marc Benioff at Salesforce.com calls it "client/service," a great name to describe this architecture. I think Marc's company will be one of the very big winners of 2004 (as it has been in 2003).

5. We're finally seeing the realization of the opportunity of the ASP. The lesson we've learned is that you can't simply deliver software as a service unless it was built from the ground up to be delivered that way.

Mitchell Kertzman
Partner
Hummer Winblad Venture Partners   Table C

Rajiv Gupta: Outsourcing, "Nike"zation

1. We will see a somewhat better IPO market but that will not offset the M&A trend. However, I think the M&A will be less consolidation and more "outsourced R&D" of the type made infamous by Cisco.

2. Large companies that did not invest in internal R&D as they should have will buy smaller companies with cutting-edge products and technologies. We will see the beginning of the "Nike"zation of the IT industry, where a large company will own branding, quality, and customers.

3. The year 2004 will see the coming of age of Web services. Linux, as the operating system to manage (provision, secure, etc.) physical components, and Web services, as the operating system (middleware) to manage application components, are going to be the basis for much of the innovation in *i*technology in this decade.

Rajiv Gupta
CTO
Confluent Software   Table D

Tim Bray: Belt-tightening, UI alternatives

1. I'm optimistic that we'll see more deals get done this year, but anyone who loosen their belt is nuts, because the economy is still very fragile and the nineties tolerance for excess and sloppy procurement culture is just totally over.

2. I think that interest in user-interface alternatives is really going to get legs in 2004; the state of the art in Web UI has been frozen for almost a decade now, and that just can't go on.

Tim Bray
CTO/Founder
Antarctica Systems Inc., Canada   Table E

Rick Ross: The Java Revolution, Google, IBM

1. Some of the remaining visionary talent at Sun will finally become so frustrated with the company's handling of Java that they will quit in disgust.

2. IBM will continue to profit hugely from the perceived safety they offer and their ability to be all things to all people.

3. [asked if he thought Microsoft would possibly buy Google in 2004] I would certainly be sad to see Google assimilated by the Redmond Cube. Google is my home page, and I'd have to change it.

4. The Java Revolution will *not* be televised, but it will make great strides forward in 2004 as developers begin to leverage the combined strength of Java Webstart, Swing, and a vastly increased installed base of compatible Java runtimes.

Rick Ross
Founder
Javalobby   Table F

Vijay Tella: Convergence, grid architecture, productivity

1. In 2004 we'll see an increasing convergence of business applications, application servers, and database tiers – driven by the need for businesses to overcome fragmented data, applications, and IT infrastructure and keep up with the increased pace of e-business.

2. This convergence, combined with the secular trend for businesses to consolidate around fewer strategic vendors, will have major implications for the enterprise software space. Vendors that get it right and deliver real solutions to customers will emerge as the big winners. Here's an example of convergence, a business integration solution that spans all three tiers of IT: ETL and data warehouse technologies in the database tier; EAI and B2Bi infrastructure in the middle tier; and data models and business processes in the applications tier.

3. Similarly, traditional business intelligence based on historical information, real-time business intelligence or business activity monitoring in the middle tier, and corporate performance management dashboards at the application tier will converge to enable a transparent, optimized enterprise. In addition, compliance will not simply be a point solution but an overarching aspect of a consolidated, integrated IT environment, combining content management in the data tier, business process management in the mid-tier, and compliance-enabled applications.

4. Another trend will be increasing the adoption of enterprise grid architecture. This will be impelled by the desire of business to drive down costs and the complexity of their IT infrastructure. Enterprise grid architectures are finally ready to deliver mission-critical performance, availability, and security in easy-to-manage commodity computing grids.

5. The final trend will focus on developer productivity. Advanced development frameworks will emerge that combine traditional, visual, declarative, and model-driven programming and make all of this accessible to a wide range of skill levels.

Vijay Tella
Chief Strategy Officer, Oracle Application Server
Oracle Corporation   Table G

> "The year 2004 for Linux will open with the 2.6 Kernel going to general availability. This is exciting because it will improve the behavior of Linux on form factors other than servers, such as workstations. I see the growth coming in horizontal scaling and in the acceptance of the Java Desktop [based] on Linux"
>
> —*Samuel J. Greenblatt* on *Linux & Web Services*
> SVP and Chief Architect for CA's Linux Technology Group

Ross is genuinely more concerned about Sun though than about Microsoft, as you can see from his various predictions in Table F.

Other Java Voices, Other Java Views

Another voice we naturally wanted to hear is that of Vijay Tella, chief strategy officer, Oracle Application Server (see Table G). We also invited the man considered by many to be "the father of XML," Sun's Jon Bosak (Table H) – whose answers naturally included a reference to his latest baby, UBL – and *Wireless Business & Technology's* editor-in-chief, Bill Ray (Table I).

'Open Source Will Become a Safe Choice'

One of the best-known "new kids on the Java block" is Marc Fleury, who more formally is the CEO of the JBoss Group. He was one of the many we consulted on the burning question of whether or not open source/Linux poses an economic threat to proprietary software.

A Final Thought for 2004

Let's give the last word though to Java advocate Rick Ross, founder of Javalobby. In response to *JDJ*'s question, "Who in your opinion is the person of the year in the *i*-technology world and why?," he stated:

> "In about a year, companies will start to see that outsourcing is not working for them as they had hoped. Companies will see that the quality of code developed offshore isn't getting any better and they will end up spending just as much to fix the code as they would have to if it was developed by their own team"
>
> —*Adam Kolawa* on *Outsourcing*
> Cofounder and CEO of Parasoft

The obvious answer to this is "The Unknown Developer," who fights and wins the technical battles without fanfare, much as "The Unknown Soldier" did. I'm tired of the idol-making and ego-stroking in this industry. It has always been the quiet, unknown developer who selects which technologies will succeed or fail and makes the world better for users everywhere.

Development managers and developers alike, *JDJ* suspects, would say "Amen" to that.

Enjoy your own 2004!

John Bosak: Open source, ebXML, Universal Business Language

1. I predict that the movement toward an open source desktop infrastructure will expand into business integration as well, with nonproprietary solutions like ebXML and UBL displacing the various vendor specifications currently competing in this space.

2. There is a growing realization that you can't let vendors control basic infrastructure standards, and I believe that this trend will continue.

Jon Bosak
Distinguished Engineer
Sun Microsystems

Table H

Bill Ray: JTWI videophones, premium messaging

1. The tech slump isn't a slump; it's a rationalization, so it will steady out.
2. Web services isn't going anywhere!
3. JTWI will be widely adopted and we'll see Java applications that aren't games on phones.
4. Psion is looking increasingly vulnerable to a takeover bid (controlling interest in Symbian) – not by Nokia, but likely someone else.
5. People will finally realize they don't want to watch movies on their phones.
6. The biggest growth will be in simple mobile services, paid for through premium-rate text and providing very simple information or entertainment.

Bill Ray
Editor-in-Chief
Wireless Business & Technology

Table I

Questions & Answers

Q: Does open source/Linux pose an economic threat to proprietary software?

Open source increasingly will become a safe choice and displace proprietary software in infrastructure. However, application software remains proprietary.

–Marc Fleury

On the contrary, I would say that proprietary software is a threat to open source. Linux and the like are going to have an increasingly difficult year as more and more patents restrict what they can do.

–Bill Ray

Does it pose a threat to overly rewarded monopolists, you mean? Of course it does, and thank goodness! Does it provide a threat to the economic foundation of a commercial software industry? No way! It will take some adapting, but the open source movement and Linux are creating new opportunities, not threatening commerce. All other things being equal, no user should ever prefer closed, proprietary products over open, standard ones.

–Rick Ross

Yes. Eric Raymond has probably been the leading thinker on the economics of OSS and conventional software, see http:// www. catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/.

–Tim Bray

No, I believe open source and Linux are the engines of a Renaissance in proprietary software – they provide the canvas on which companies will build (paint, so to speak) proprietary applications. Having a common commodity and inexpensive medium did not inhibit individual expression, it enabled it. Yes, open source does make technical, product, or service stagnation very uncomfortable. The sooner we rediscover how to innovate, the faster we will appreciate the value of open source for economic opportunity. Until then it will be a perceived threat (at least to those who are slow to leverage it).

–Rajiv Gupta

Q: Is the tech slump over – or will 2004 bring continued belt-tightening in the industry?

The "slump" is a low point relative to the superfluous and poorly directed spending that occurred during the "boom" when John Q. Public decided that NASDAQ would be more fun and rewarding than Las Vegas. I doubt that the public will become enamored of putting their savings into such high-risk investments again during 2004, so spending is likely to match the more critically determined needs of home and business consumers. Ask yourself: Is your computer fast enough? Does it do what you need it to? If the answer is yes, then you're not likely to be on a tear with new spending in 2004. Why should you? I don't plan to be.

–Rick Ross

I'm optimistic that we'll see more deals this year, but anyone who loosens their belt is nuts, because the economy is still very fragile and the nineties tolerance for excess and sloppy procurement culture is just totally over.

–Tim Bray

Q: Who, in your opinion, is the "Person of the Year" in the i-technology world and why?

Linus [Torvalds], for keeping the Linux gang pointed more or less in the same direction and working efficiently.

–Tim Bray

Irving Wladawsky-Berger of IBM for giving "enterprise-respectability" to Linux. If I can add another recommendation it would be to the folks at HP for their support of E-Speak, which by most accounts fathered Web services.

–Rajiv Gupta

Q: What single advance or improvement would you most like to see in the i-technology world?

Speed and ubiquity in always-connected broadband mobility. I don't want to be tied to the desktop, and I need a device that works everywhere and gives me high-speed Internet access in a form factor I can tolerate.

–Rick Ross

Java Licensing and Browser Wars

Sun's Java leadership days are over. The Sun "Java Cartel Process" (JCP) will collapse. The "Free Java Era" will end and Sun will start charging for Java offering a SCO-like Java IP licensing scheme. And the browser war is back. This time it's not about HTML but about next-gen markup languages such as XUL, SVG, and XForms.
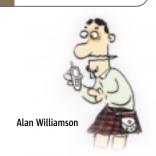
–Gerald Bauer
XUL Pioneer

...And How About that *Other* Technology?

I'd like to see Microsoft put some real investment into improving the user experience – something they talk about a lot – in particular, into the software environment where most people spend most of their time, namely the Web browser. It seems really questionable that there isn't going to be an IE7 and that we're stuck with the antiquated IE6 code (which is really the IE4 code internally) until Longhorn arrives.

–Tim Bray
Antarctica Systems, co-inventor of XML

Calling Dick Tracy...
Calling Dick Tracy

Alan Williamson

All change on the *JDJ* front, eh? What's going on? It's all gone topsy-turvy! "Alan at the back, a whole host of new faces, and a new front cover." Don't panic; it's all under control. *JDJ* has undergone a major renewal process, taking us into version 3. We do this every so often so we don't get complacent and to ensure that we are always providing you with information that best serves you. The office of editor is an important one and when we rev another release, we like to appoint a new editor-in-chief to control the next wave. I'm very proud and pleased to announce Joe Ottinger as the next editor-in-chief of *JDJ*. Joe has an impeccable pedigree and is a very lively, passionate, and highly intelligent individual who will steer *JDJ* through the next stages of its growth.

The team we have assembled at *JDJ* is worthy of special mention. Our latest addition is Kirk Pepperdine, who takes on the role of "enterprise editor." Kirk is someone we've all known for a long time; we have been fortunate enough to have him write for *JDJ* a number of times. In this issue, we also have Joe Winchester and his new editorial section, "Desktop Java," which will be focusing on the major developments Java is making on the client side of the equation.

an urge to play with J2ME, first ask a teenager what he or she thinks would be "cool" and code accordingly. Although one startling fact emerged: for the first time in history, ringtones of the (UK) Top 40 pop chart are outselling the actual Top 40 chart! Not sure what that says about the current pop industry, but it can't be good.

Dick Tracy Watch

Java is popping up on phones and wireless devices all over the shop. Barely a week goes by when there isn't some announcement being made regarding the number of Java-enabled units available. It's a shame the development market isn't growing accordingly. Have you noticed the distinct lack of free Java MIDlets? Everyone is out to profit, to hopefully claim their piece of the "China" effect (you know, sell a small item to every Chinese citizen and you've made a billion customers). I only wish we would see in the MIDlet world what we had seen in the applet days: lots of free downloads. But mobile phones aside, did you see that in Tokyo, just before Christmas, Aplix announced a Java-enabled wristwatch (www.gizmodo.com/archives/011091.php)? Looks very interesting and probably is a hot contender for this year's JavaOne cool giveaway.

> " Java is popping up on phones and wireless devices all over the shop...
>
> ## It's a shame the development market isn't growing accordingly"

I haven't cut all ties with *JDJ*, mind you. I'll be building out and managing the online side of *JDJ* and creating an environment that we can all contribute to. In addition, I'll be profiling and interviewing key individuals in the Java space as we look at the major inroads Java is making in our lives. Plenty of work still to do and we are all looking forward to getting our teeth into it.

Location, Location, Location

So this is the back page. Look at all this space – how wonderful. In this space I'll be looking at the world of Java and profiling some of the more interesting tidbits that have caught my attention in the past month – generally the cool stuff that doesn't require a whole page of editorial or a product review, just a paragraph or two. Stuff to put faith back in the world of Java.

Ringtones

Driving one Saturday afternoon I heard a report on BBC Radio regarding major wireless operators and where they were making most of their revenue. No great surprise to hear that the teenage market is the model customer for the likes of Vodafone and T-Mobile. Gone are the days of pandering to the stuffy old businessman or salesman. The next time you have

BitTorrent

Go on, admit it, which file sharing service do you use? Kazza? Morpheus? One that's sneaking around the outskirts is BitTorrent (BT) (http://bitconjurer.org/BitTorrent/). BT operates differently from the others in the sense that there is no one centralized server and it's true peer-to-peer. The more people sharing a file, the faster the download for others, as each person downloading also uploads a small share of the same file. Red Hat has begun to ship their latest releases using BT, which takes a lot of the bandwidth away from Red Hat, while at the same time offering users a much faster download rate. I'm also looking at this for our purposes here at SYS-CON as a potential way to speed up the digital edition PDFs. Because the technology is open source, there are plenty of clients popping up. One that I've settled on is Azureus (http://azureus.sourceforge.net/). It took me a wee while to realize that this was in actual fact a Java SWT application. It's very slick, operates quickly, and downloads really fast. The UI is rich with many features and cool widgets. It's a good demonstration of an SWT application in action.

On that note, I'm off. Need to chase up a rumor I heard that Sun is looking to license .NET from Microsoft! Stranger things have been known to happen.

Alan Williamson, when not answering your e-mails and working on the next issue of *JDJ*, heads up a small team dubbed the "Thunderbirds of the Java industry," providing on- and offsite rescue for Java projects in trouble. For more information visit www.n-ary.com. You can also read his blog: http://alan.blog-city.com.

alan@sys-con.com