

JAVA™ DEVELOPER'S JOURNAL

The World's Leading Java Resource

August 2000 Volume: 5 Issue: 8

JAVA DEVELOPERS JOURNAL.COM

XML DevCon FALL 2000 November 12-15, 2000

Announcing... **Wireless DevCon** December 3-5, 2000

From the Editor
by Sean Rhody pg. 5

JDJ Guest Editorial
Readers' Choice Awards
by Alan Williamson pg. 7

IMHO
J2EE for the Real World
by David Skok pg. 8

CORBA Corner
by Jon Siegel pg. 22


Interview
Jeremy Allaire
of Allaire Corporation pg. 28

Product Review
Java 2 Platform Standard Edition 1.3/ Enterprise Edition 1.2.1
by Jim Milbery pg. 38

App Server History
by Phil Costa pg. 50

From the Publisher
by Fuat Kircaali pg. 86

Editorial
by Ajit Sagar pg. 101



See page 85

2000 READERS' CHOICE AWARDS

FREE Pass Inside!
JavaCON 2000
See full program inside for details
DON'T MISS IT!
The largest gathering of Java Developers this fall!
Sept. 24-27

Feature: <Extending> Application Servers  **Ajit Sagar**
Now Playing: XML, in a crucial role in application server suites **14**

VisualAge Repository: Debugging and Unit Testing Server-Side Web Applications  **Jin Li**
Boost productivity with incremental compilation and hot linking **34**

Reusable Components: Component Assembly Using a Visual Development Environment  **Jeff Chapman**
Accelerate the development process from concept to working Web application **44**

E-Business: Java Application Servers: Buyer Be Aware  **Anita Osterhaug**
'HTML guru wanted in Oakland' to build a complex B2B commerce site' **56**

Feature: The Power of Polymorphism  **Jim Barnabee**
Build polymorphic architectures and lower the cost of developing good software **62**

Feature: The Use of JNDI in Enterprise Java APIs **Ian Moraes**
A common framework for locating enterprise computing components **72**

EJB Home: J2EE and WML: Holding a Royal Flush  **Jason Westra**
Will you hold the winning hand? **80**

Progress

www.sonicmq.com/ad11.htm

Protoview

www.protoview.com

Segue

www.seguel.com

SEAN RHODY, EDITOR-IN-CHIEF



The Next Big Thing

"J2EE is to Java what SQL was to databases." That's a direct quote from one of my conversations with software vendors during JavaOne. And I tend to agree with the statement. I spent most of the show talking with people regarding their products, their visions and their strategies. It was clear to me that J2EE is a standard with enough meat behind it to generate adoption by many companies as the clear design for their products.

What also quickly became apparent was that there's a certain degree of parity in J2EE. Looking at the vendors involved, you could go from booth to booth (and I did) and hear the same message from each: "We're J2EE compliant." I had long talks with IBM, BEA, SilverStream, GemStone and several others, and the message was pretty much the same regarding compliance. One might be missing JMS or some other minor part of the standard, but they all provide servlets, JSP, EJB and the other necessary technologies needed to build a Web application using Java.

The similarities to the SQL standard are strong. SQL is a standard by which you can ask a database for information. Every relational database vendor supports it – you could go as far as to say it's not a database if it doesn't support SQL. In the same fashion, J2EE is defining how we'll develop Java applications; you could say that without J2EE it's not an application server.

In some ways this is like having 20 or 30 identical choices to make: If they all support J2EE, what's the real difference between one and the next one? But again, like SQL, J2EE provides enough wiggle room for vendors to differentiate themselves from each other. Part of this will be in the area of high availability and reliability as vendors provide different support for clustering. Some vendors like BEA cluster at the machine level; others such as GemStone cluster at the VM level. I heard some reblings from the vendors that over time VMs aren't as stable as we'd like and need to be restarted. I'm not sure I believe this, but I'd be interested in getting your feedback on the issue.

Tuning their servers and improving the scalability and availability of the products is important, but I don't think that's enough of a differentiator. The real way I think vendors are going to differentiate themselves is through the additional services they offer to complement the basic application server technology. The pace at which our industry moves is incredible and there's constant pressure to deliver an application in 90 days. The B2B market wants quick solutions, but they need to be customizable and at the same time flexible enough to accommodate constant changes in requirements.

I'm already seeing the signs of this differentiation. BEA is offering a commerce package built on top of a component framework. GemStone has actually developed an entire J2EE e-commerce application and placed it in open source as a resource for all developers. Persistence has developed a page-caching engine that works with dynamic, personalized content.

There's no clear winner at the moment, not even a clear direction. I'd like to see offerings that perform large amounts of business functionality – such as an auction engine or a logistics system – come as packages built on the application server. Sybase, for example, is providing financial and telecom services that leverage their EAServer. I'd like to see the same for NetMarkets. Just as the focus in the NetMarket has shifted from exchange technology to service integration, so do we need a shift from the nuts and bolts of the application server business into a more business-solution-focused offering. I know I'm asking a lot. But the people I work for these days all ask the same of me: they want it fast...they want it in a package...and the package should be easy to reconfigure when the business rules change.

So I want that type of offering from the application server vendors. I want to be able to buy an application server and be 80% done with the project instead of a quarter of the way done. And of course I want it soon...say, in 90 days. ☺

sean@sys-con.com

AUTHOR BIO

Sean Rhody is editor-in-chief of Java Developer's Journal. He is also a respected industry expert and a consultant with a leading Internet service company.

JAVA DEVELOPER'S JOURNAL

EDITORIAL ADVISORY BOARD

TED COOMBS, BILL DUNLAP, DAVID GEE, MICHEL GERIN,
ARTHUR VAN HOFF, JOHN OLSON, GEORGE PAOLINI, KIM POLESE,
SEAN RHODY, RICK ROSS, AJIT SAGAR, RICHARD SOLEY, ALAN WILLIAMSON

EDITOR-IN-CHIEF: SEAN RHODY
EXECUTIVE EDITOR: M'LOU PINKHAM
ART DIRECTOR: ALEX BOTERO
MANAGING EDITOR: CHERYL VAN SISE
ASSOCIATE EDITOR: NANCY VALENTINE
EDITORIAL CONSULTANT: SCOTT DAVISON
TECHNICAL EDITOR: BAHADIR KARUV
PRODUCT REVIEW EDITOR: ED ZEBROWSKI
INDUSTRY NEWS EDITOR: ALAN WILLIAMSON
E-COMMERCE EDITOR: AJIT SAGAR

WRITERS IN THIS ISSUE

JIM BARNABEE, JEFF CHAPMAN, PHIL COSTA, GERARDO DADA,
JIN LI, JIM MILBERY, IAN MORAES, ANITA OSTERHAUG, AJIT SAGAR,
JON SIEGEL, DAVID SKOK, JASON WESTRA, ALAN WILLIAMSON

SUBSCRIPTIONS

FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT

SUBSCRIPTION HOTLINE: 800 513-7111

COVER PRICE: \$4.99/ISSUE

DOMESTIC: \$49/YR. (12 ISSUES) CANADA/MEXICO: \$69/YR.
OVERSEAS: BASIC SUBSCRIPTION PRICE PLUS AIRMAIL POSTAGE
(U.S. BANKS OR MONEY ORDERS). BACK ISSUES: \$12 EACH

PRESIDENT AND CEO: FUAT A. KIRCAALI
VICE PRESIDENT, PRODUCTION: JIM MORGAN
VICE PRESIDENT, MARKETING: CARMEN GONZALEZ
GROUP PUBLISHER: JEREMY GEELAN
CHIEF FINANCIAL OFFICER: ELI HOROWITZ
ADVERTISING ACCOUNT MANAGERS: ROBYN FORMA
MEGAN RING
JDSTORE.COM: AMANDA MOSKOWITZ
ADVERTISING ASSISTANT: CHRISTINE RUSSELL
ADVERTISING INTERI: ALISON NOVICK
GRAPHIC DESIGNER: ABRAHAM ADDO
GRAPHIC DESIGN INTERI: AARATHI VENKATARAMAN
WEBMASTER: ROBERT DIAMOND
WEB SERVICES CONSULTANT: BRUNO Y. DECAUDIN
WEB DESIGNER: STEPHEN KILMURRAY
CUSTOMER SERVICE: ELLEN MOSKOWITZ

EDITORIAL OFFICES

SYS-CON PUBLICATIONS, INC.
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9600
SUBSCRIBE@SYS-CON.COM

JAVA DEVELOPER'S JOURNAL (ISSN#1087-6944)
is published monthly (12 times a year) for \$49.00 by
SYS-CON Publications, Inc., 135 Chestnut Ridge Road, Montvale, NJ 07645.
Periodicals Postage rates are paid at
Montvale, NJ 07645 and additional mailing offices.
POSTMASTER: Send address changes to:
JAVA DEVELOPER'S JOURNAL, SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

© COPYRIGHT

Copyright © 2000 by SYS-CON Publications, Inc. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or by any
means, electronic or mechanical, including photocopy or any information storage and
retrieval system, without written permission. For promotional reprints, contact reprint
coordinator, SYS-CON Publications, Inc., reserves the right to revise, republish and
authorize its readers to use the articles submitted for publication.

WORLDWIDE DISTRIBUTION BY CURTIS CIRCULATION COMPANY

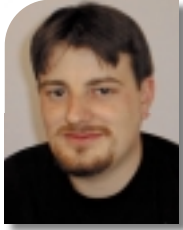
730 RIVER ROAD, NEW MILFORD NJ 07646-3048 PHONE: 201 634-7400

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.



Together Soft

www.togethersoft.com



Readers' Choice Awards? What Do They Really Mean?

Awards and opinion polls are funny things, don't you think? In general, you read them with interest and suddenly you find yourself believing what they're saying. You then go and tell others about them, disseminating the information as the gospel truth. How many times have you found yourself saying to someone, "Yes, it's the most popular product"? As if saying it gives you justification for using the beast. Have you ever read the wee small writing at the bottom of any of these polls? If you haven't, you should. It may shock you to discover just how few people they "polled" for this definitive guide. How can such a small group of people truly represent the mass population?

The same goes for readers' awards polls. When the claim is that such-and-such and thus-and-so are the most popular products, yet it's backed up with only 400 or so results, you'd be more than suspicious. However, some leading magazines attempt to pass this off on you, hoping you won't notice. Sadly, **JDJ** will have to stand apart from the crowd. For when it runs its annual Readers' Choice Awards, we actually get over 20% of our readers casting their votes. More than 20,000 of you took the time this year to vote for a wide range of products, and we'd like to thank you for making this readers' award poll meaningful.



And the Winner Is . . .

So with the number 20,000 as "proof," let's have a quick look at some of the winners.

The most popular category has to be "Best IDE." This is one piece of software that becomes a very close friend of developers as we sit hour after hour deploying our solutions. For the second year running, IBM's VisualAge takes the top place, with my favorite IDE, Metrowerks' CodeWarrior, coming in at position 7. Interesting to see that all those that made the decision to go to VisualAge are still happy with that decision.

The winner of the "Best Application Server" went to Apple's WebObjects, for the second year running. It was good to see Apache's JServ come up the ranks from last year's poll to grab the third slot. And good to see that many of you rate the free alternative above many of the big commercial brands.

In the "Best Application" category there was a major change of heart from you all. Last year's runner-up, Jasmine, came in at position 12, with Together/J pulling the majority of the votes for the top place.

An interesting category that I was keeping a particular eye on was the "Best Database Product." I wanted to discover just where the heavyweights would place – it was no great surprise to see Oracle 8i taking the number one slot, outstripping Sybase's votes by more than double.

Sometimes there's one winner that stands head and shoulders above the rest, and an example of this can be found in the "Best Book" category in which *Java in a Nutshell* by David Flanagan took the majority of the votes. Taking this in context, removing all the specialized Java books, *Java in a Nutshell* was obviously the choice of the majority of developers as being a great all-around Java book that deserves some real estate on your bookshelf.

In the "Most Innovative Java Product" category, VisualAge took the honors for the second year running, with JDeveloper coming in second. It's interesting when the tool you know and love is the one that picks up the majority of the votes.

A new category this year was the "Best Virtual Machine," with Sun's HotSpot picking up the majority of the votes to beat ChaiVM.

These awards are indicative of what you, the readers, think, and as can be seen, the categories were varied and thoughtfully voted on. If you're popping along to JavaCon 2000 in Santa Clara, California, September 24–27, stop by and meet some of the vendors that have won your votes. Let them know what you think. That's what they're there for.

To this end, we put a closed lid on this year's awards, but be sure to stick around: we'll need you again next year! ☺

alan@sys-con.com

AUTHOR BIO

Alan Williamson is CEO of n-ary consulting Ltd., the first pure Java company in the United Kingdom. The firm, which specializes solely in Java at the server side, has offices in Scotland, England and Australia. Alan is the author of two Java servlet books, and contributed to the Servlet API. He has a Web site at www.n-ary.com.

No
Magic
www.magic-draw.com

J2EE for the Real World

A major battle is brewing for the huge e-business platform market

WRITTEN BY DAVID SKOK



The advent of J2EE (Java 2 Platform, Enterprise Edition), a market-revolutionizing industry specification that standardizes the way that application servers work, sets the stage for a collision of two markets: content management/personalization and back-end data access/transactions.

It also becomes a driving force as organizations move to create powerful new e-business applications that bring together all forms of electronic interaction for customers, trading partners and employees.

The J2EE standard has been uniformly adopted by almost every application server vendor and is being enthusiastically adopted as the standard for Web development by much of corporate America. It unifies the programming interfaces or APIs to the most important areas in an application server. Just as SQL did for relational databases, J2EE is creating tidal waves in the Web application marketplace. Once database vendors adopted support for the SQL standard, no packaged vendors would dream of writing their own database. In the world of Web applications the same is now true of application servers. It makes no sense to write a Web application without using a standard application server. Several packages were written prior to application servers and their creators were forced to write their own proprietary application server functionality. These vendors find themselves in a tough position, faced with the need to either rewrite their entire functionality or find their market share shrinking as buyers shun their proprietary approaches.

The J2EE standard also brings several other advantages:

1. Application developers with J2EE expertise are widely available.
2. Additional components for specific functionality (e.g., e-commerce) can be bought from third parties and plugged into the server.
3. A wide range of tools are now tailored for J2EE use, including popular products such as Macromedia Dreamweaver.
4. Expertise and skills learned while building applications can be reused on all other application projects.

As J2EE becomes the core technology for real-world Web applications, we're witnessing unprecedented demand for J2EE-based e-business solutions such as portals, e-commerce sites, B2B commerce solutions and so on. This has become a top priority for IT.

In the past many of these solutions have been built on top of IRM (Internet Relationship Management) products whose primary specialization has been content management and personalization. For early Web sites this was adequate, as they were mostly serving up static content. Once the basic site is up and running, however, the demand quickly shifts to more data-driven applications in which customers can place orders, see inventory, book travel, track packages and more. This requires the ability to develop highly scalable, enterprise-

class Web applications that access existing corporate data and applications and provide high-end transactional capabilities.

In particular, almost every application server vendor – to meet the demand from their clients for more prebuilt functionality – has begun developing e-business solutions that include IRM functionality based on top of their application servers.

Prior to the J2EE standard, Web software companies, including IRM vendors, wrote their own servers to support their applications, frequently spending as much as 70% of their energy on that task. These were proprietary, with weak architectures that lacked important features like distributed objects, transactions, standardized data access and message queuing. Furthermore, they provided proprietary scripting languages that weren't up to the task of writing serious enterprise applications. Today no software company in its right mind would write its own proprietary server. A company would simply write to the J2EE standard and support the leading J2EE servers in the market.

Because of the availability of J2EE standardized servers, there's a backlash at customer sites against the proprietary server architectures. Organizations standardizing on J2EE want to use this architecture for all their e-business applications; that way, they invest in learning only one skill set that can be used across all projects – and they're easily able to find developers who already know the J2EE standard. They're also able to use a much wider range of development tools as products such as Rational Rose and Macromedia Dreamweaver are adapted to support the J2EE standard. Furthermore, they can purchase from third-party vendors add-on components and prebuilt software modules that comply with the J2EE standard.

As the content management/personalization vendors collide with the application server vendors, we see a major battle brewing. At stake is the huge e-business platform market. In one corner are the IRM content management and personalization products built on a proprietary architecture with weak scripting capabilities. In the other are the application server vendors who are moving into e-business solutions, leveraging their strong architecture.

The obvious move for the existing IRM vendors is to rewrite their products to be J2EE compliant. Companies such as Vignette are describing multiphase plans to do something like this. This is a tough task though, and may well take years to accomplish correctly; during this time it will be close to impossible to simultaneously evolve their solutions functionality at a rapid pace. Probable outcome: loss of momentum and market share.

Several application server vendors are already pursuing the strategy of adding e-business solutions to their product lines and moving into the complete e-business platform space. Not all will succeed, as the solutions skill set is different from the systems skill set that most possess. Those that do will take market share away from the IRM vendors and may have a shot at the brass ring.



KLGroup

www.klgroup.com/power

Real World E-Business Solutions and J2EE

Savvy businesses have realized the need to create powerful new Web sites that bring together all forms of electronic interaction for customers, trading partners and employees. These e-business Web sites need to recognize the context of individual visitors and adaptively bring together all the resources needed to help them find what they're looking for and transact business. These next-generation e-business Web sites make it easier for people to navigate around a vast sea of information and to personalize the way the site works for them. They also offer powerful business applications and transactions.

Next-generation e-business Web sites require a multitude of important capabilities. They must possess:

- **The ability to handle all forms of interaction:** Initially, Web interactions were restricted to users visiting a site via a Web browser. But wireless devices connected to the Web, such as WAP/WML-enabled cell phones, are expected to eclipse the PC population (200 million PCs) by the year 2003, growing rapidly to over 500 million devices. There's also a need to allow for B2B transactions that take place directly with trading partners' applications using XML.
- **Fresh, constantly updated content:** This is required to drive traffic to a site and keep it interesting. Content also drives transactions (e.g., equity research causes people to trade stocks and rich graphical product catalogs drive consumers to make purchases). Discussion boards help create communities, which in turn drive traffic. Web sites that share knowledge bases with customers and employees greatly improve the flow of critical information needed for things like customer self-service.
- **Richly functional Web applications:** In addition to content capabilities, organizations need to provide richly functional Web applications. These are different for each particular industry. For example, insurance industry users will want to view insurance policies and claims, whereas in the travel industry users will want to book travel. These applications need the capacity to handle large numbers of business transactions.
- **Connectivity:** Web applications need to be able to access legacy mainframe applications, packaged applications (SAP R/3, PeopleSoft) and relational databases, as this is where existing functionality lives. The existing functionality needs to be repurposed with new Web interfaces. Multiple systems need to appear seamlessly integrated so the customer/visitor sees all the relevant information in one place.
- **XML capability:** These applications also need to be able to communicate directly with those using XML. This enables seamless electronic integration with trading partners and trading exchanges. It also eliminates paper and human intervention, both of which are costly. Processing a paper purchase order typically costs around \$75 versus around 80¢ for an electronic PO.
- **Personalization:** Personalization has become one of the most important tools in the online business world. Just as personalized service in the physical world creates satisfied loyal customers, it's also critical to provide personalized service in the electronic world. As soon as visitors reach a site, observing their behavior allows organizations to learn about them and start personalizing their experience. As visitors become more interested and as trust develops, they will tell the organization more about themselves. This information can then be used to further personalize the site. As visitors experience the benefits of increasingly relevant information, their trust evolves rapidly to higher levels and it becomes increasingly likely that they'll transact business. As they start entering transactions, it's possible to immediately recommend other prod-

ucts that work together with what they're purchasing. After they've become customers, it's possible to use information from internal systems to cross-sell and up-sell additional products and services.

- **"My Yahoo"-style personalization:** As a longer-term business relationship builds, customers and other site visitors will want to clearly define their own personalization. This involves allowing them the flexibility to lay out their own pages, assembling components of information in a way that makes sense to them. Each component needs to be customizable to allow them to choose the information they wish to see. Examples here would be financial clients specifying which stocks should be on their watch list and what types of equities research they'd like to see.
- **E-mail:** E-mail provides a powerful way to reach visitors after they've left a site, to promote additional products and services. Like content on the site, e-mail messages need to be personalized to make them highly relevant; otherwise they can be considered offensive.

J2EE-based application servers are highly flexible platforms for building richly functional Web applications. They enable developers to rapidly evolve new functionality for Web sites, and provide the scalability, reliability and security that organizations need. By embracing industry standards they help organizations avoid vendor lock-in. However, as seen from the requirements above, J2EE-based application servers are

missing many of the pieces required for e-business, such as content management, XML, personalization and other critical e-business features. Developers can develop these pieces on their own, but with the pressure to deliver e-business solutions fast – ahead of the competition – the time and cost of building from scratch becomes unmanageable. On the other hand, while prepackaged solutions have the advantage of fast time-to-market, they often lack the functionality, flexibility and scalability that organizations need.

The dilemma developers face is exacerbated by confusion in the e-business platform market. The prepackaged solution vendors that promise fast time-to-market will face a challenge from the traditional application server players who promise scalable, reliable, standards-based solu-

tions – and vice versa.

As the market evolves, hybrids of these two worlds will emerge (or collide). In fact, evidence of this can be seen by more recent e-business platforms that offer a combination of flexibility, fast time-to-market and a standards-based architecture. As such, the vendors that will ultimately win this battle will provide a "best of both worlds" solution – delivering not only the fast time-to-market that e-business demands but also the flexibility to extend functionality by leveraging the underlying power of J2EE.

Conclusion

This is all good news for developers. With a platform that provides core features including personalization, content management, a component framework and an extensive library of components, developers get the fast time-to-market, flexibility and rich features they need, on top of a J2EE standard application server foundation. Using such a platform, developers can focus on the solution, not the technology – quickly delivering the robust applications their e-business initiative demands. ☺

AUTHOR BIO

David Skok is chairman and founder of SilverStream Software, Inc.

dskok@silverstream.com

The dilemma developers face is exacerbated by confusion in the e-business platform market

Gemstone

www.gemstone.com/welcome

Info

www.cloud

rmix

scape.com

NOW PLAYING:

XML, in a crucial role in application server suites

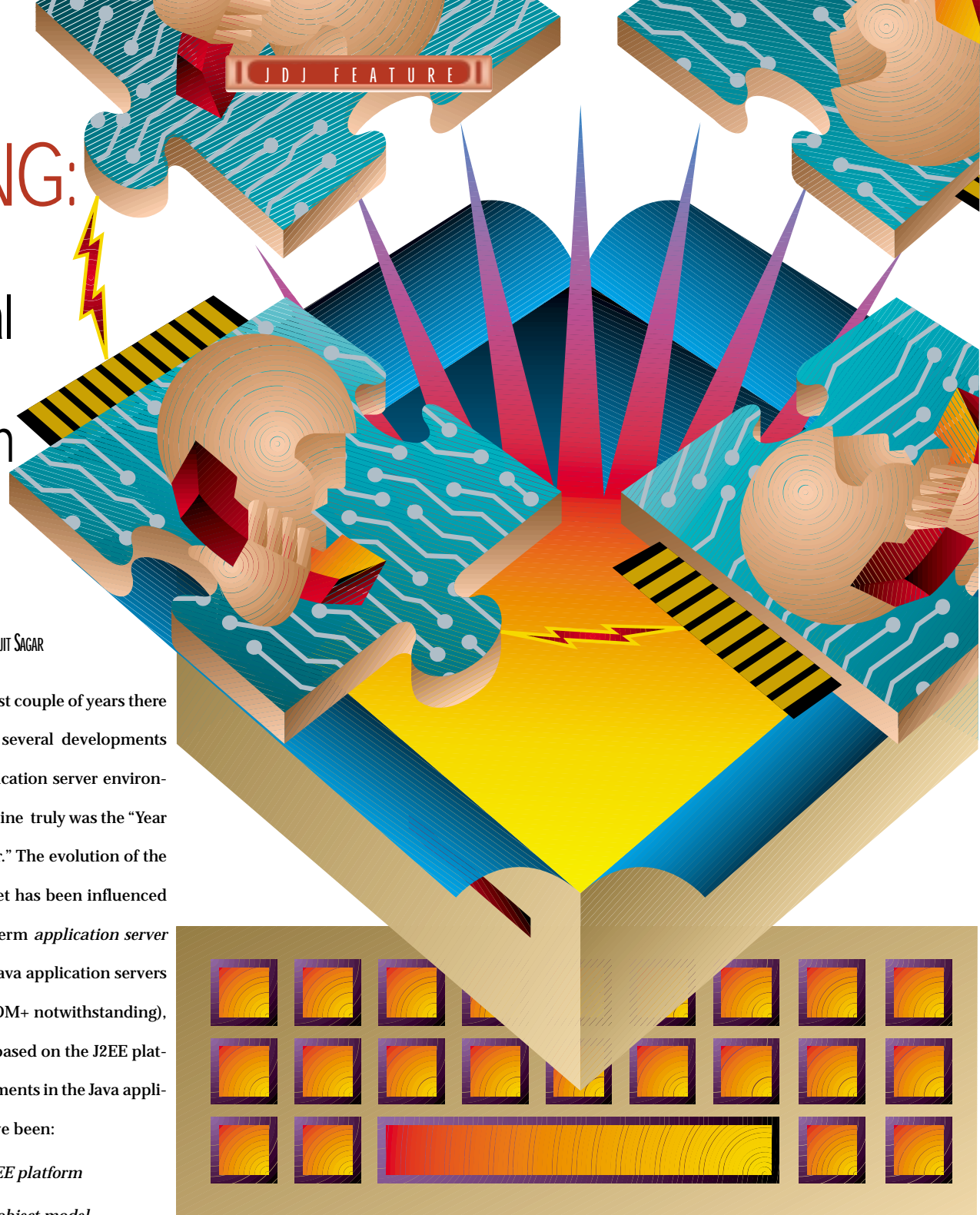
WRITTEN BY AJIT SAGAR

During the last couple of years there have been several developments in the application server environment. Nineteen ninety-nine truly was the “Year of the Application Server.” The evolution of the application server market has been influenced by several factors. The term *application server* almost always refers to Java application servers (Microsoft’s MTS and COM+ notwithstanding), i.e., application servers based on the J2EE platform. The main developments in the Java application server market have been:

- *Stabilization of the J2EE platform*
- *Acceptance of the EJB object model*
- *Maturing of application server vendor offerings*

Application server vendors have expanded their product lines by means of acquisition, innovation and conglomeration. Application servers now address application development and integration issues in enterprise-level e-business applications. In that sense vendors have expanded beyond their realm of runtime execution environments to e-business solution environments.

< e X t e n d i n g >
APPLICATION SERVERS



BEA

www.bea.com

The terms *XML* or *XML support* are invariably found wherever e-business and enterprise applications are mentioned. XML (eXtensible Markup Language), the new technology that's rapidly changing the way Web developers design and build their e-business applications, is an open standard for defining and sharing data on the World Wide Web. XML finds application in several e-business applications such as supply chain management, customer relationship management and content management by enabling proprietary systems to more easily exchange structured information. In the last year XML technology has gained a lot of ground and momentum.

As application server vendors expand their offerings into e-business solution suites, XML plays a crucial part in enabling them to interchange data between the various components of the suite. This article examines current trends in the application server market and the role of XML in application server suites.

Application Server Overview

An application server, by definition, is a computer server that serves applications. More precisely, it serves up application services. Its main purpose is to reduce the workload of applications by taking over the responsibility of mundane activities involved in executing the application and making the application's services available to external modules in a reliable manner. I'll take a stab at defining an application server as follows:

- An application server is a computer program that resides on a server in a distributed network whose main function is to provide the business logic for an application program.
- An application server provides a customizable and flexible execution environment for hosting business logic components, thus providing distributed services and integrity for application execution.
- An application server provides an execution environment that decouples front-end clients from back-end data access. The execution environment is supported by an infrastructure that enables integration among different applications. Application servers enable this integration by offering software components that can be used to create business logic for an enterprise application. The supporting infrastructure may include architectural frameworks such as messaging systems, transactional managers and database accessors.

Application servers came into existence because of a need in the market for flexible, robust, extensible and standards-based enterprise applications developed at Internet speeds. The Internet is the most powerful phenomenon driving application development and deployment today. Internet applications are shared among multitudes of parties participating in e-business transactions. This results in the need for standard architectures and frameworks that allow application hosting. Application servers provide the execution environment for Internet applications. Vendors in the application server market add value to the equation by taking over the burden of application hosting and offering commodified products that enable organizations to concentrate their resources on building the applications themselves. In some ways the application server market is moving toward the space currently occupied by operating systems. Operating systems are developed and maintained by third-party vendors. Companies use OS services to develop applications in their business niche. Similarly, application server vendors can provide third-party services for use by distributed application developers.

The benefits to an application development vendor of using a third-party application server are:

- Better product focus
- Abstraction
- Indirection
- Application interoperability
- Better resource utilization
- External support and maintenance

Java application servers are a by-product of Java's increasing presence in server-side middleware and the definition of Java Enterprise APIs by Sun Microsystems in collaboration with its industry partners. Java Enterprise APIs define enterprise-level services for server-side deployments. As

described earlier, application servers started appearing in the market when the concept of multitiered computing became popular. These servers provided a hosting environment for middleware components. However, before the stabilization of Java Enterprise APIs, the definition of middleware components for a ubiquitous software platform was not uniform across operating systems. Hence, application servers were operation-system specific, each providing middleware services in a proprietary way, making portability and reuse of the components a difficult task. Emergence of the Java Enterprise APIs has enabled definition of a standard architecture for middleware components that comprises business objects. This architecture clearly defines well-formed interfaces between the application server's object containers and the objects or components themselves. In Java this is made possible by:

1. A standards object model (EJB) for designing business objects
2. Uniform APIs for accessing business objects (remote interfaces via RMI)
3. Container APIs for interacting with vendors' mechanisms to access system resources (EJB home interfaces)
4. APIs for finding business objects (JNDI)
5. Standard means of accessing these components through a distributed protocol (Java Servlets, RMI)
6. Standard APIs for connecting to back-office data sources (JMS/JavaMail, JDBC, JTS)
7. Secure access and data interchange (Java security API)

These form the components for a basic Java application server, as illustrated in Figure 1.

Developing E-Business Solutions

E-business drives the Internet economy, which is dependent on technology for modeling real-world businesses. Products offered by various vendors now need to provide functionality to realize different components of an e-business application including procurement services, business transaction services, security, order processing and purchases. Most enterprises look to vendors for value-added solutions that can provide these services in the form of out-of-the-box product suites.

Until recently, application server vendors provided the technical infrastructure to build enterprise applications. Their focus has been on the middle tiers of a distributed application. This middle tier can be split into two layers – the presentation layer that deals with getting input from the client tier and the business logic layer that is responsible for processing the data and providing connectivity to back-office systems such as ERP systems and databases. The application server product cycle is following the same trend as the operating systems product cycle did several years ago. Operating systems provided an abstraction on top of system services. Now operating system vendors have been reduced to the few that survived the OS wars. Application servers are the “operating systems” of e-business. However, they're more complex because they span several hardware and software platforms.

The application server market is at a stage where a large part of the paradigm has already solidified. Standard Web and Internet protocols

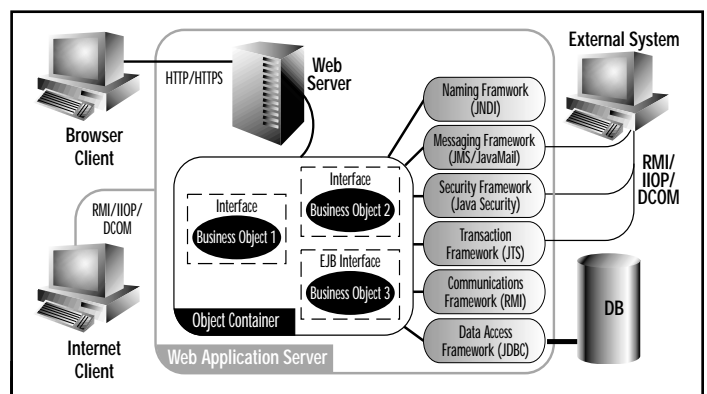


FIGURE 1 Basic components of a Web application server

NovaMedia

www.novamedia.com

(HTTP/TCP), object models (EJB/COM/CORBA) and distributed communication protocols (RMI/IIOP) have matured and are being accepted globally. In the next few years the application server market will mature to a level similar to that of the OS market today. The market is already looking toward application server vendors to see what type of integration alternatives they offer to fit in with the other building blocks of an e-business application. How do they integrate with content management systems? How do application servers work across different computing platforms? How do they offer commerce services like procurement and buying and selling services? How do they connect to logistics providers? How do application servers talk to legacy systems?

Figure 2 illustrates some of the other layers required to complete an e-business application.

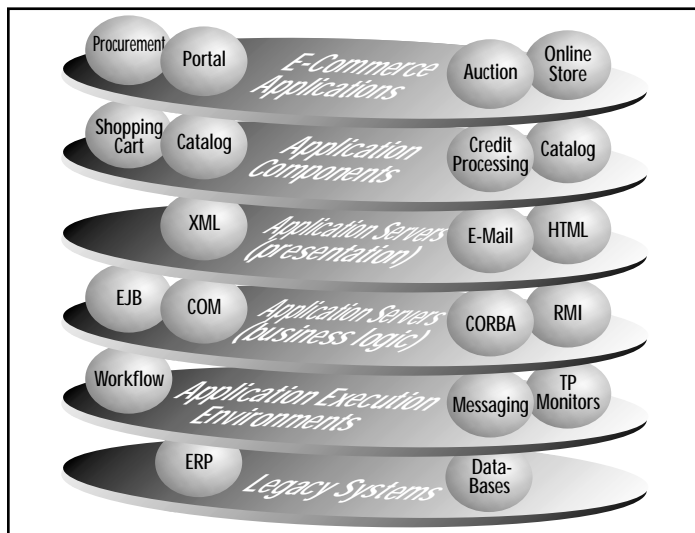


FIGURE 2 Layers of an e-business application

Application servers provide development and execution environments for business applications. This is the core from which they started. However, complete solutions require integration with other environments such as ERP systems. This integration falls under the umbrella of enterprise application integration (EAI). The EAI space has been evolving too. Some vendors have concentrated on solving the EAI issues for integrating legacy applications into the new Internet paradigm. Organizations building e-business solutions prefer buying third-party products to meet time-to-market pressures. They count on application server vendors to integrate with these products. This has led to partnerships and acquisitions that couple app servers with integration servers. The combined product suite is often called an e-business server. E-business servers offer a variety of services including CRM, content management, personalization and trading community services. These suites offer development environments in several facets of e-business development. Figure 3 illustrates these development environments.

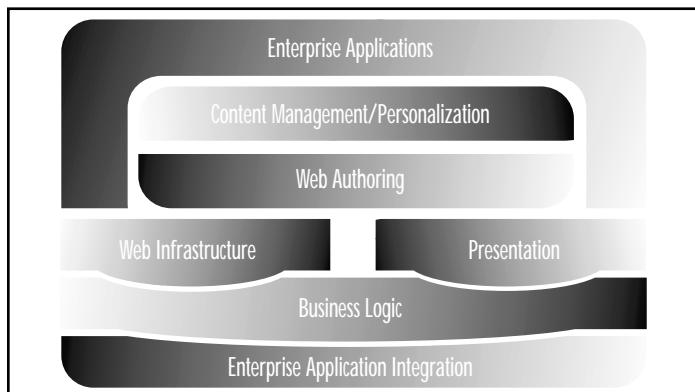


FIGURE 3 Development environments in e-business servers

Enter XML

So where can XML be leveraged for e-business application development? The answers are:

- Wherever data is transported between distributed processes
- Wherever data is presented to the user

Figure 4 illustrates the areas where XML can be used to support integration between different e-business application servers and between application components. The current market trend for application server vendors is to claim that they have “XML integration support.” What exactly does this mean? XML can be used in several different application development components. Integrating with XML means that app servers should support the following:

- The presentation logic should be able to serve up XML to a variety of clients.
- Business logic should be able to extract XML, process it internally, package it for external processing and feed back the results to the client.
- External applications must be able to create and manage dynamic, secure business relationships across the Web using XML as the exchange format.
- Enterprise business systems should be able to handle XML-based transactions.
- Application servers should integrate with technologies including DOM, SAX, DTD, XSD, XPointer, XLink, XPath, XSL and XSLT.

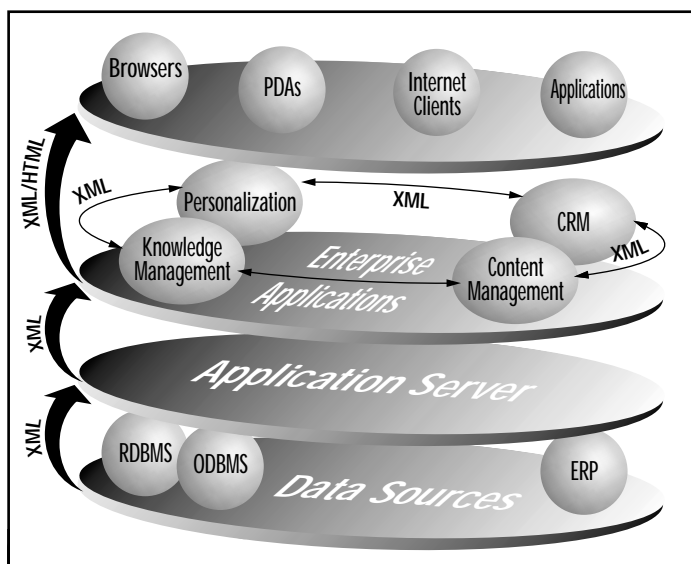


FIGURE 4 Use of XML in an e-business application infrastructure

One of the first uses of XML in conjunction with J2EE-based application servers was to create deployment descriptor documents for Enterprise JavaBeans. Since XML is an elegant format for expressing data, it’s ideal for expressing configuration of the transaction and security properties or server-side Java components.

XML Servers

I’ve recently seen the terms *XML application server*, *XML-oriented application server*, *XML integration server* and *XML B2B server* in different places. In fact, if you go to Yahoo’s site and look for XML servers, these terms will show up in the results of your search. So there are a couple of questions to be answered before you go any further. What is an XML application server? Frankly, I don’t know. Going by a traditional application server definition, an XML application server should be able to serve up XML documents in a robust, secure, scalable environment. Thus, if you add XML processing tools to any standard application server, you should be able to have an XML application server. In Java terms this means that, on the client side, customized Servlets and JSPs should be able to accept, process and generate XML. On the other side, it should

Compuware

www.compuware.com/numega

Visualize

www.visualizeinc.com

Advanced Data

[www.advanced-
data.com](http://www.advanced-data.com)

be possible to have business objects that extract information from the XML document that comes from the client and package it into some message format using JavaMail or JMS to send it to back-office systems. Other business objects should get XML from back-office sources, process it and serve it up to the client.

However, there's an independent XML server market evolving outside the realm of standard application servers. These XML servers address integration issues with legacy systems. Vendors are entering this application space from different origins. Some of the areas in which XML servers contribute to enterprise applications are:

- Web catalogs
- Knowledge management
- Supply-chain integration
- EAI, data integration
- Data aggregation
- Business workflow
- Content management and publishing

Nowadays several application server vendors are including an XML server in their product suites. This allows them to extend traditional app server offerings of a robust and secure execution environment to a product suite that enables integration with EIS (Enterprise Information Sources). Examples of some of these product suites are given below. This is not an exhaustive list, but merely some examples of XML application server combos available in the market.

BLUESTONE TOTAL-E-B2B

Total-e-B2B allows automation of your supply chain, enhancement of logistics operations and extension of e-business relationships with customers, suppliers and business partners. It uses XML technology to enable cross-platform communication, solving logistical and back-end business-to-business challenges. Bluestone offers this XML server in addition to a suite of products – Total-e-B2C for CRM and Total-e-Server, which is Bluestone's Java application server (formerly known as Sapphire/Web).

SILVERSTREAM'S XCOMMERCE

The xCommerce server is designed specifically to permit business analysts and software engineers to rapidly enable their proprietary systems for XML integration, map the data flows of those systems to other XML-enabled applications, and manage the runtime environment through which integrated applications interoperate. xCommerce enables B2B connections between Internet commerce applications, Web portals and vendor packages. SilverStream's corresponding application server offering is their SilverStream application server, which is J2EE compliant.

These are just two examples of combinations of XML servers and Java application servers. The XML server market itself has several products to offer in the XML server market, such as eXcelon Corp.'s B2B Integration Server, WebMethod's B2B and Microsoft's BizTalk Server.

Conclusion

The application server market is stabilizing with the acceptance of a few distributed object models and distributed platform technologies. The XML server market is rapidly evolving and offering solutions in the e-business space. Alliances between these two markets are taking place today and will continue to do so in the future. Vendors are offering "total e-solutions" that include bundling a Java application server with an XML server into one product suite. The integration between these products is not necessarily tightly coupled or even complete. Other vendors are partnering with complementary products to offer complete e-business solutions. The focus of application server vendors seems to be shifting from product providers to e-business solution providers. ☪

ajit@sys-con.com

Allaire

www.allaire.com/download

The Future of CORBA: A Look into OMG's Crystal Ball

What do the next few years hold in store for OMG's technology?

WRITTEN BY
JON SIEGEL



The extensive suite of Object Management Group (OMG) standards will, ultimately, unify computing from analysis and design through development, deployment, runtime and support.

OMG is an open, member-driven organization, and future directions emerge from the work of its nearly 800 members. Thus, like any other process that tracks developments in our rapidly moving industry, it's hard to see more than a few years into the future. It takes from about 14 to 20 months to create an OMG specification from the time OMG's members publish their requirements as an RFP (request for proposal), which is probably the fastest open consensus-building process in the computer industry. In this article I'll use this lead time to look a year or two into OMG's future. Market trends are another good indicator of trends so I'll include a dose of them too.

Predictions

Taking all of OMG's technology into consideration, the group has come up with six predictions. If our crystal ball is right, watch for:

1. **New UML release:** A major new release of the Unified Modeling Language (UML), OMG's showcase standard for analysis and design. The role of OMG's Analysis and Design standards – UML, the Meta-Object Facility (the MOF), XML Metadata Interchange (XMI) and the Common Warehouse Metamodel (CWM) – will also expand as more and more enterprises embark on large-scale application integration projects.

2. **Glue standards:** New "glue" standards at the boundaries between OMG's technology areas – Analysis and Design, core CORBA, and Domain facilities – will unify an environment in which applications flow smoothly and automatically from analysis and design, coding and deployment, into support.

3. **Domain Technology expansion:** OMG's Domain Technology program will expand its penetration into vertical-market computing as products based on OMG standards reach the marketplace in finance, healthcare, manufacturing and other areas.

4. **CORBA marches on:** CORBA platform technology will continue to embrace emerging enterprise technologies.

5. **Core technology comes to market:** Recent extensions to the CORBA core technology – in particular, asynchronous and messaging mode invocations and the CORBA Component Model (CCM) – will reach the marketplace and mature.

6. **COBRA in specialized distributed systems:** CORBA will play an increasing role in distributed embedded and real-time systems.

In the remainder of this article I'll discuss these predictions in more detail, starting with the new UML release.

OMG Standards in Analysis and Design

Starting with the UML and branching out, OMG now maintains a suite of standards that unifies both application and data modeling under a common hierarchy. The first extension beyond UML was the Meta-Object Facility (MOF), which standardized both a meta-model and its representation, providing the key foundation elements. It also standardized a repository, confirming that its meta-model was useful in code as well as in concept.

The next extension defined XML Metadata Interchange (XMI). Enterprises invest in a standard model because it is useful in more than one application or tool, but this requires transporting the model from one place to another. XMI is the transport format that allows models



Tidestone

www.tidestone.com

to be moved from tool to repository, repository to tool, or from one tool to another. Expressed in XML, XMI combines three industry standards: XML, UML and the MOF.

The most recent addition is the Common Warehouse Metamodel (CWM). Enterprises mine data warehouses for new business opportunities. (Think of this as more ways to earn the money that goes into your paycheck!) But conflicting models in different warehouses within the same company make analysis difficult. For example, engineering and manufacturing use a model that emphasizes how the company's products are put together, while marketing and sales use a model that emphasizes what they do and why customers want to buy them. The two models may even use different terms for the same concepts. CWM overcomes these difficulties without forcing either side to use a model that doesn't meet its needs.

Why are we devoting this much space to modeling standards in a programmer's magazine? As Java matures, it's being used more and more to build huge servers that have to be designed before they get coded. Even if you won't ever design anything yourself using UML, it's likely you'll be asked to code something that someone else did (either that or someone else who can code from a design will get that contract or raise instead of you). We think familiarity with analysis and design techniques and tools is a handy skill for every enterprise programmer.

OMG Analysis and Design Directions

OMG members are currently drafting an RFP for a new major release, UML 2.0. Unlike point releases (UML 1.1, 1.2 and so forth), which can only fix bugs and clarify ambiguities, major releases contain new features. Even the RFP, the requirements document for the new release, involves a lot of work. OMG members expect to issue the RFP for UML 2.0 before the end of 2000; when it does, it will appear on OMG's "Work in Progress" Web page, www.omg.org/schedule, where you can follow it through the process.

Extending the modeling arena, members are working on a standard covering XMI production of XML schemas. This new work will unify XMI and the MOF with W3C's emerging standard for schema representations in XML. This illustrates two recurring themes in OMG's work:

1. Members try to integrate all of the significant work in an area to keep the group's standards useful.
2. OMG "works well with others," estab-

lishing liaison relationships with other standards groups and using their specifications wherever they apply.

SOFTWARE PROCESS ENGINEERING

A new direction is Software Process Engineering (SPE). Long ignored by standards efforts, SPE includes software project management but goes beyond it, encompassing team organization and culture, choice of technology and other real-world aspects of producing applications. One SPE RFP is in process now; a look through the document reveals that more will follow until the space is fully populated with standards. The group at OMG that pushes this effort calls itself the Process Working Group (PWG); if your company is an OMG member and you want to follow it, e-mail request@omg.org and ask to be put on the e-mail list proc@omg.org.



Business Object Initiative: "Glue Standards"

The biggest area of new work is in what you might call "glue standards." (No, people don't call it that at OMG, but I think it's a handy descriptive term.)

Glue standards unify OMG technology at the boundaries. Now that OMG specifications provide so much help within each area of application development – analysis and design, infrastructure, standard objects – it's even more noticeable how much work you have to do to jump over the boundary from one stage of your project to the next. For years people have wondered why it was such hard work to transition from a work product in one language to the next – from UML to OMG IDL and from IDL to Java or C++.

With standards well established within each area, it's finally possible to standardize the glue between them, and OMG members are putting a lot of effort into this. The overall effort has been named the Business Object Initiative (BOI). The first stage of the BOI will produce four base standards. The first is ready and currently under vote, the next two are in process and the fourth is an RFP draft waiting to be issued. Here are some details:

UML PROFILE FOR CORBA

UML is a comprehensive language and its models can apply to many infrastructure and development environments besides CORBA. (Some people think that OMG hasn't noticed this yet. We have!) This flexibility is a great enabler, but it also makes for awkward moments when you try to implement a UML model, since nothing stops you, when you construct your model, from using concepts that don't carry over into your target development environment. To streamline this transition for CORBA projects (our favorite target environment here at OMG), the first specification in the BOI is a UML profile for CORBA. By restricting your UML space to this profile, a UML tool will guide its user to produce a model that moves smoothly into a CORBA development. This helps whether your transition will be manual or automated (as the BOI anticipates). This process has a few procedural votes left before it becomes an official specification, but you can read the almost-specification now at www.omg.org/cgi-bin/doc?ad/00-02-02 and www.omg.org/cgi-bin/doc?ad/00-06-50.

UML TEXTUAL NOTATION

The only "language" for UML right now is graphical. (XMI is great at transporting a model from one place to another, but it doesn't represent the model in a manipulable way.) This effort will produce a human-readable and possibly machine-editable text-based representation for UML.

UML PROFILE FOR EDOC

Enterprise Distributed Object Computing (EDOC) – large applications with high throughput and reliability requirements, along with integrated legacy applications spread over the enterprise – represents a big market for UML and CORBA. This RFP will produce a profile for UML aimed specifically at this market and its unique requirements. Its completion will set the stage for the fourth RFP, described next.

CapeClear

www.capeclear.com

MAPPING OF THE UML PROFILE FOR EDOC TO CORBA

This has to wait for the EDOC UML profile so submitters know what they need to do. When this mapping is complete, you'll be able to model your EDOC application in UML and move the model into a CORBA implementation, experiencing no pain at the boundaries.

The target CORBA environment of the BOI isn't the version you're coding in now – it's CORBA 3, with all the features of the CCM. The support for a tiered architecture, homes for component types, and built-in scalability, transactionality and security combine to make CORBA 3 the ideal environment for enterprise servers. By standardizing the way it provides these services, CORBA 3 enables automation of the transition from design to code.

OTHER INITIATIVES

There's one more RFP working now that will expand the modeling area: UML Profile for EAI (Enterprise Application Integration). EAI has been receiving a lot of attention recently, and this profile will make it easier to model your EAI project in UML.

There is also work underway, not yet visible, to create UML profiles for real-time and minimal (i.e., embedded) systems. We'll touch on these one more time in our last prediction.

OMG's Domain Technology Program

Since the first meeting of OMG's Domain Technology Committee in January 1996, domain-specific activities have grown as large as the platform-side activities like CORBA and UML standardization and are still the fastest-growing part of the organization. All nine Domain Task Forces are working on new standards and have more planned. The nine are :

- Business Objects
- Finance/Insurance
- Electronic Commerce
- Transportation
- Manufacturing
- Healthcare (sometimes called CORBAmed)
- Telecommunications
- Utilities
- Life Science Research

This topic is big enough to get its own article, and we're already working on it. You can follow all of this at www.omg.org/schedule.

Since we're talking futures, we also need to mention groups that aren't quite task forces yet but are expected to emerge. New groups are starting to coa-

lesce in these areas: Space Science, Retail, Customer Relationship Management, Digital Content Management and Semiconductors. If you work in one of these areas and would like to get involved at the ground floor, send e-mail to info@omg.org for more information.

CORBA Platform Technology Embraces Emerging Enterprise Interoperability Modes

People are always searching for something. In the Middle Ages it was the philosopher's stone and the Holy Grail. In enterprise interoperability it's the silver bullet. Each new technology comes in accompanied by a promise that it does everything better and will replace all current technology. ("Instead of n technologies to support, you'll only have one!") After a few years, things settle down with the new technology taking its place alongside all of the previous stuff, doing the things it does better and leaving everything else to the rest. (Instead of n technologies to support, you end up with $n+1$!)

CORBA's forte is integration. As new technologies come (and never seem to go), CORBA extends a welcoming arm and integrates them into the rest of the distributed environment. When everything is sorted out, at least our $n+1+1+1\dots$ technologies are well integrated.

The new technology in the news now is XML. OMG has already used XML to good advantage in XMI and the CWM; in addition, configuration files in CCM are generated in XML. A new specification, due to complete around the time this article goes to press, will define a mapping from XML to OMG IDL, translating XML documents into IDL types so they can be sent as parameters to CORBA object invocations. This remedies XML's lack of semantics for invocations, transactions and security. Watch for a **CORBA Corner** highlighting this new specification in a few months.

Consolidation of New CORBA Core Technologies

ORB vendors – including OMG members as well as nonmembers – are now in implementation mode, assembling and implementing the CORBA 3 technology adoptions that extended OMG into new territory: the Asynchronous Messaging Specification including network quality of service control, the CCM and the Persistent State Service are the most obvious ones, but there are about half a dozen others as well. These

new specifications will allow software vendors and CORBA users to take better advantage of CORBA within the enterprise, and in interenterprise computing. Early implementations should be available on the market by the time you read this, although some specifications may take another generation to mature. The CCM contains so many advances that it's worth the short wait.

Even more extensions to CORBA are either underway or contemplated, including wireless transport (in cooperation with the Telecommunications DTF), parallel processing, multicast and others. Check the Work in Progress Web page (already referenced) under ORBOS for details.

CORBA in Real-Time and Embedded Systems

OMG members have adopted specifications for both real-time and embedded (officially termed *minimal*) CORBA and are now extending these in several directions. One new standard configuration, not yet even an RFP, is nicknamed *microCORBA*, a configuration with a footprint smaller even than minimal CORBA. OMG's first workshop on distributed real-time and embedded systems, held in Virginia in July of this year, demonstrated the industry's interest. We'll write up the workshop in a future column.

An area not represented in RFP yet but likely to appear is modeling of real-time and embedded systems. Members are already working on draft RFPs in this area; you'll see them on the Work in Progress page as soon as they issue.

Summary

That's as far ahead as my crystal ball can predict, at least on the platform side of OMG. Most of the DTFs have issued road maps organizing their thoughts on future work. Combined with draft RFPs and work in progress, these documents give us insight into the IT future of Telecommunications, Manufacturing, Transportation, Healthcare, Banking, Insurance, Electronic Commerce and others. We'll review these in a future column.

In the meantime, if your company is interested in participating in any of OMG's standards efforts, you can find out more at our Web site. We've already cited the Work in Progress page; for membership information (since you have to be a member to contribute to OMG specifications), surf to www.omg.org/membership or e-mail me directly at siegel@omg.org.

siegel@omg.org

AUTHOR BIO

Jon Siegel, the Object Management Group's director of technology transfer, also writes articles and presents tutorials and seminars about CORBA. His new book, *CORBA 3 Fundamentals and Programming*, has just been published by John Wiley and Sons.

Hot Dispatch

www.hotdispatch.com



Interview...with **Jeremy Allaire**

COFOUNDER AND CTO OF ALLAIRE CORPORATION, INC.

AN INTERVIEW BY AIT SAGAR



JDJ: *Allaire has recently been in an acquisition and partnership mode. Can you give us a brief history of the events over the last couple of years and the rationale behind these decisions?*

Allaire: One of the primary reasons for going public over a year ago was to create a base of assets that could be leveraged into mergers and acquisitions. We saw a number of areas for expansion for our platform, all of which underlined a long-term strategy for becoming one of the leading Internet business platforms. In 1999 we undertook three acquisitions. The first, BrightTiger Technologies, makers of advanced scalability and Web systems management technology, underpinned our efforts to bolster our Enterprise-level application server offering. The last two, LiveSoftware and Valto Systems, helped to accelerate Allaire's entry into the Java server marketplace, and have formed the foundation for our next-generation products. As you know, both LiveSoftware and Valto were 100% focused on pure, standards-based Java architectures, including JSP, servlets, EJB, JTA and JMS. Unlike a lot of other players in the Java server field who were carrying forward their own proprietary Java offerings, we saw that the end of 1999 was the time for a pure-play, standards offering.

JDJ: *It seems to me that this rapid evolution has confused the marketplace. Is there a uniform message Allaire wants to send out to the Web community?*

Allaire: A lot has happened with Allaire in the last year. We've evolved from being a leading provider of tools and application servers to supplying a comprehensive Internet software platform covering core server infrastructure, packaged applications as well as development and productivity tools. If there is a uniform message for the Web community, it's that Allaire intends to be a dominant provider of Internet software platforms, enabling any organization to successfully build their business on the Web. We intend to pro-

vide top-to-bottom platform infrastructure, as well as a wide range of horizontal packaged applications necessary to running an Internet business.

JDJ: *Going back to my first question, how does this message map to the products you're offering?*

Allaire: In our recently released Allaire Technology Roadmap, we discuss the customer requirements for e-business software infrastructure, and then map this to our existing products and platform, as well as provide a roadmap for future Allaire offerings. Today we're providing a very robust set of application server products covering both front-tier and back-end application server requirements. This goes from free products for small dynamic sites up through enterprise offerings, including distributed transaction and message queuing infrastructure. On the packaged applications front we released Allaire Spectra 1.0 in December, providing a comprehensive suite of modules for content management, e-commerce and customer relationship management on the Web. We also announced the acquisition of Open Sesame, an early pioneer and leader in the customer profiling, analysis and personalization space.

JDJ: *Allaire started out with site-building software that has now evolved into an industry-strength application server. How would you define an application server in today's market?*

Allaire: Again, we discuss this in great detail in our Technology Roadmap. It's our belief that the application server portion of e-business software infrastructure is rapidly becoming something of an "operating platform" for all Web applications, replacing the role that was once provided by classic middleware and network operating systems. Functionally, application servers really provide four key areas of functionality: Core Services, such as clustering, security, session management and logging; Integration Services, such as connectivity to Internet protocols, ERP sys-

tems, database systems, directories and so on; Web Application Logic Services, providing a dynamic page environment that is used for scripting and is generally the basis for about 80% of applications built; and, finally, Business Logic Services, providing a container or hosting environment for complex business logic, executing in a distributed, transactable fashion. This will certainly be the minimum requirements for any offering into 2000 and 2001.

JDJ: *With consolidations and mergers, where do you think the application server market is going? Where does Allaire stand in that market?*

Allaire: I think the foregoing outlines our thinking in some detail. Clearly Allaire is in a very strong position, with some 25% of the commercial market according to surveys by IDC. In terms of revenue from application servers, we're right at the top of the pack. The market will continue to consolidate into 2001, and will be driven by two major forces. One is the obvious and increasing role of standards in this infrastructure, with J2EE as the most notable driver, and second is recognition that application servers are true platforms, and that the winners in this space will need to provide an offering that spans every tier of project, skill set and environment. When a larger corporation, say a Ford, or someone buys into a standard server platform, they'll look for that to scale from small-scale departmental sites and applications up into the core back-end for their e-commerce transactions. Our focus on horizontal coverage of the marketplace, combined with products that go very deep in terms of enterprise capabilities, certainly sets us up well to compete.

JDJ: *Nowadays, to get wide acceptance in the computer industry, companies try to adhere to technology standards. Does Allaire plan to standardize any of its technologies so as to get a larger audience?*

Allaire: Broad adoption of any platform requires a unique combination of proprietary innovation and open standards. This

has certainly been the case in the Internet world. For Allaire, this means building and supplying infrastructure based on Internet and industry standards, innovating beyond standards in territories that aren't developed, and in turn collaborating to ensure that that innovation eventually contributes to open standards efforts. Our efforts in the servlet and JSP community are indicative. The JRun team has consistently extended what's possible with server-side Java, and has been aggressively contributing that work to the Sun Community Process. Likewise, with XML protocols Allaire is committed to a similar effort.

JDJ: *What is your roadmap for the coming year?*

Allaire: We've got a lot of things in the works. First off, we're shipping JRun 3.0, a comprehensive Java application server offering, including full support for J2EE standards, and support for distributed transactions and message queuing, as well as JRun Studio, our first Java-focused IDE product. Later in the year we expect to ship Spectra 2.0, which will add new modules for customer intelligence, personalization and merchandising, as well as expand the core capabilities of the 1.0 offering. We'll also be delivering Tron, the code name for a new B2B Integration Server product based on XML middleware. Into 2001 we're planning to deliver next-generation versions of ColdFusion and JRun through an integrated server code-named Pharaoh, as well as a new offering code-named Harvest, providing comprehensive Web systems management capabilities for managing large farms of Web application servers.

JDJ: *Does Allaire plan to get into application design or do you plan to always be application enablers?*

Allaire: I assume by this you mean actually getting into the application development business through a consulting organization. This is not a big focus for the company, though we are expanding our services offerings to better ensure the success of our enterprise customers.

Persistence

www.persistence.com

JDJ: *Since some of our JDJ readers may not be familiar with Allaire Corporation, could you provide an overview of what Allaire does, how your company reached where it is now and what your plans are for the future?*

Allaire: Allaire is a leading supplier of Internet software and services for companies building their business on the Web. Our Internet business platform is in use by over 10,000 corporations and over 500,000 developers worldwide. There are three major pillars in our platform: application servers, packaged applications and visual tools. Our application servers, ColdFusion and JRun, are widely adopted for the creation and delivery of Web applications. We also provide a suite of packaged applications under the Allaire Spectra product line, providing solutions for content management, e-commerce and customer relationship management on the Web. In addition, we supply a range of visual tools that ensure the productivity and empowerment of every employee in a Web organization.

JDJ: *How does all this relate to Java technologies?*

Allaire: Over the past year Allaire has made massive investments in server-side Java infrastructure through acquisitions of leading products and technologies as well as through our own internal R&D. The JRun 3.0 offering that's in late beta right now provides companies with an extremely comprehensive suite of software based entirely around the J2EE standards. This product line will be marketed and priced aggressively for broad adoption, and will form the foundation for a range of next-generation products.

JDJ: *What is your relationship with Sun and the Java community?*

Allaire: We're very involved in the major standards efforts for Java, including a strong focus on J2EE services. In fact, the JRun team has been at the forefront of defining these standards over the last few years.

JDJ: *How does your entry into the Java middle tier affect the company's direction?*

Allaire: The focus on component and transaction middleware is a very important evolution in the company's efforts to provide a complete Internet business platform. It's also a critical foundation for our packaged commerce software, as well as for new products in the B2B Integration space.

JDJ: *Don't Sun's technologies for the client, such as JSP and servlets, overlap with your own, such as CFML? After all, both are mechanisms for producing dynamic HTML.*

Allaire: CFML, JSP and ASP all provide mechanisms for delivering dynamic pages to browsers. I wouldn't put servlets proper in this category, as servlets are really a Java-focused alternative to CGI or NSAPI/ISAPI, and really don't provide a scripting and page template environment that's necessary for Web applications. Currently, CFML provides the highest level of abstraction and therefore the highest degree of productivity for developing interactive applications. Both JSP and ASP are still at the object-scripting level, and don't yet provide the same kind of productivity advantage, though they're ages ahead of writing server-side interactivity logic using Java or C. As you may be aware, Allaire has been one of the most active proponents for the standardization of tag-based scripting within JSP, contributing to the Taglib architecture and working actively on reference implementations as well as next-generation architectures. As we move ColdFusion to a J2EE foundation, you'll also see us move CFML to sit as an abstraction on top of JSP and servlets.

JDJ: *Until June, Allaire didn't have much representation in the Java world. Then you went and bought JRun. What prompted you to do so? How is that merger working out?*

Allaire: It was very clear that Allaire had an opportunity to be a leading provider of standards-based server-side Java infrastructure, and we saw all of our products moving in this direction. LiveSoftware had clearly emerged as one of the dominant suppliers in the JSP and servlet space, with over 80,000 developers using their products. This was a great way to buy into a large community building around standards, and to leverage that into a broader set of products focused on server-side Java.

JDJ: *Your recent acquisition of Valto Systems broadcasts the message that you want to compete with the big boys of Java enterprise computing, such as BEA Systems, Sun and IBM. How do you plan to gain entry into an already defined market?*

Allaire: We expect to become a major supplier in the enterprise tier of the Java application server market, and I think you'll see a very different approach from Allaire

in pursuing that space. First, our entire offering is based 100% on Java, and 100% on J2EE implemented services. Unlike many of our competitors who have existing C++, Java and CORBA-based systems that are migrating to J2EE, everything we've done has been built from the ground up on Java2. This means that we've got an exceptionally lightweight, clean-room implementation that comes in at around a one MB memory footprint. We're going to leverage that into ISV and OEM customers, who are increasingly looking for embedded J2EE services. We're also going to continue the Allaire tradition of pricing and packaging our platforms for mass adoption, so I think from a developer perspective the Allaire J2EE offering will be the most competitive in terms of price/performance in the marketplace.

JDJ: *I've heard that the ColdFusion Application Server can't handle high-volume transactions. Is this true? If so, how are you addressing the issue in your migration to the Java middle tier?*

Allaire: In 1998 that was certainly true. In late 1998 and throughout 1999 we delivered ColdFusion Enterprise, our first enterprise-level application server offering. This included all of the components you'd expect at this tier – advanced scalability and clustering services, native database connectivity, CORBA and Java support, and a lot of server tuning, caching and configuration options. With this offering we were able to support dozens of large dot-com customers handling millions or tens of millions per day, and had a couple of the top 10 holiday e-commerce sites deployed entirely on our platform. In addition, our move into the transaction middleware space will ensure fail-safe deployment and reliability for customers implementing large-scale transactional systems.

JDJ: *Your ColdFusion Studio product provides an IDE for ColdFusion applications. Do you plan to provide an IDE for your Java products?*

Allaire: With JRun 3.0 we're introducing JRun Studio, our first foray into the server-side Java IDE space. With the Pharaoh release of our application server, we'll be expanding the EJB-focused functionality in our overall visual tools suite.

JDJ: *What other areas of the Java market do you plan to step into? Are you looking at other partnerships/mergers?*

Allaire: There are lots of partnering opportunities, certainly, whether in the testing tools space or in vertical applications, among other things. Our latest acquisition, OpenSesame, provides a J2EE-based customer profiling, analysis and personalization server that will be incorporated into our suite of packaged applications.

JDJ: *What is Eijpt? I noticed your Java strategy revolves around the name Egypt. Any particular reason?*

Allaire: Eijpt was the name of Valto Systems' EJB server offering. This technology from Valto Systems will serve as an underpinning for a lot of our future products and architecture, so that's part of it. We also liked the fact that we could come up with lots of interesting code-names surrounding ancient civilization.

JDJ: *Would you say that ColdFusion is a direct competitor of JSP? Is there room for both? If so, could you give examples of the kinds of applications that could leverage both these technologies?*

Allaire: Again, today, you really would choose one or the other for the Web application logic layer. Going forward, with CFML built on a Java foundation, it will be incorporated into the same runtime environment as JSP. What it will really come down to for developers is what level of abstraction they want or need to work at. CFML is simply the fastest way to develop dynamic Web applications, and we will carry this tradition forward into the standards-based community.

JDJ: *For Java developers not familiar with your products, what would be a good place to start incorporating them into their existing applications?*

Allaire: Any Web-centric or browser-based application would be a good candidate for using JRun.

JDJ: *How can our readers start playing around with some of your new products?*

Allaire: JRun 3.0 beta is available for download from beta.allaire.com/jrun30. Allaire Spectra, our packaged system, as well as ColdFusion Enterprise are both available for download from our public Web site, www.allaire.com. ☺

Ajit Sagar, a frequent contributor to JDJ and ColdFusion Developer's Journal, is also editor-in-chief of XML-Journal.

VisiComp

www.visicomp.com/jdj7

Macro

www.macromedia.com

Media

media.com/ultradev

Debugging and Unit Testing Server-Side Web Applications

How to boost your productivity with incremental compilation and hot linking

WRITTEN BY
JIM LI



Java-based technologies such as servlet and JavaServer Pages enable the effective development of component-based, server-side Web applications. Although servlets can be self-contained programs, it's more common that JavaBeans handle the business logic, JSP components handle the presentation logic and servlets handle the HTTP protocol interaction flows.

Servlet and JSP components should be tested individually to ensure that they conform to the functional specifications. More important, they should be tested after they've been integrated to form a server-side Web application. This article describes the development scenario for a typical server-side Web application and outlines the advantages of interactive debugging and iterative unit testing. It then demonstrates how debugging and unit testing can be accomplished using the IBM VisualAge for Java software development product.

Typical Server-Side Web Application Scenario

Our sample scenario involves the development of a Web-based employee telephone directory for a fictitious company, Acme Inc. Using a Web browser, employees can look up information on another employee by name, department number and so on. The application performs the search based on the input criteria and presents the results on a new Web page. Figure 1 depicts the application flow and architecture.

Our Web application uses a typical three-tier architecture. An HTML-based client user interface is served by a Web server and hosted inside a Web browser running on the end user's computer. Java, servlet and JSP code is executed on an application server, while a data access request is delegated to a database server. A simple HTML file, PhoneBookInput.html, is used as the search criteria input page. When the user pushes the Search button, the request is

passed to a servlet, PhoneBookServlet, for processing. This servlet uses JavaBeans to perform the lookup in a DB2 database via the appropriate JDBC calls and stores the found information in a JavaBean called *Employees*. The PhoneBookOutput.jsp file is then called from the servlet. This JSP file extracts information from the *Employees* bean in order to present the search results.

Need for Debugging and Unit Testing

Debugging and unit testing are part of an iterative development cycle in the software engineering process. The interpretive and dynamic nature of HTML and Java make the two tasks critical.

Servlets can be executed with the help of *servletrunner*, a utility available in the Java Servlet Development Kit (JSDK). Often, to test out a particular function in a servlet, we need to involve the associated input and output HTML and JSP components. Combined with the limitations of *servletrunner*, this approach is neither practical nor effective for rapid application development. A more common ap-

proach is to test servlets using a Web server with a servlet engine plug-in.

To meet the requirements of testing JSP files, we need a JSP processor so that JSP files are first page-compiled into servlets, then executed. You've probably encountered a typical setup for servlet/JSP development: a standard Java IDE and an external Web server with servlet/JSP support. It's more productive, however, to have a RAD environment with built-in support for interactive debugging and iterative unit testing of both servlets and JSP code.

To perform the debugging and unit-testing tasks effectively, we need the ability to (1) trace both our Java and our JSP code, (2) examine the application execution information, (3) make some changes to the code and (4) continue with the current execution without restarting from the beginning.

Debugging and Unit Testing with VisualAge for Java

The Professional and Enterprise editions of VisualAge for Java, version 3.02, incorporate the WebSphere Test

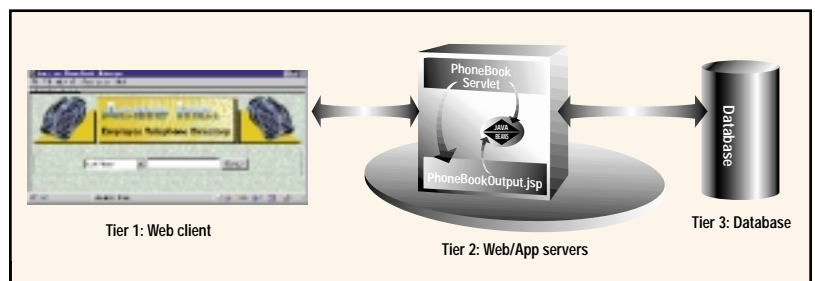


FIGURE 1 Sample application flow and architecture

Cerebellum

www.cerebellumsoftware.com

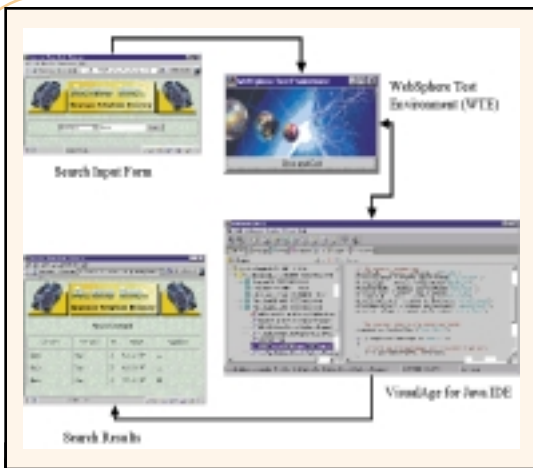


FIGURE 2 Launching the WTE and testing JSP files and servlets

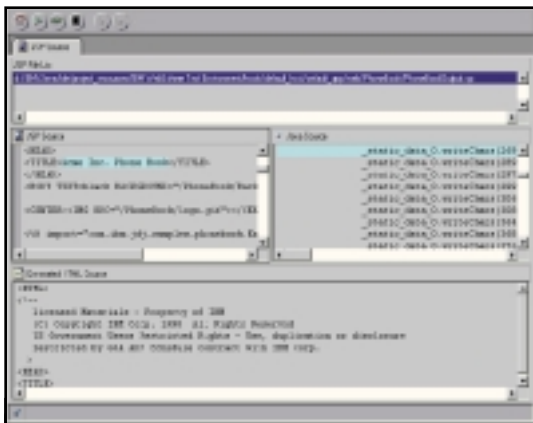


FIGURE 3 JSP Execution Monitor

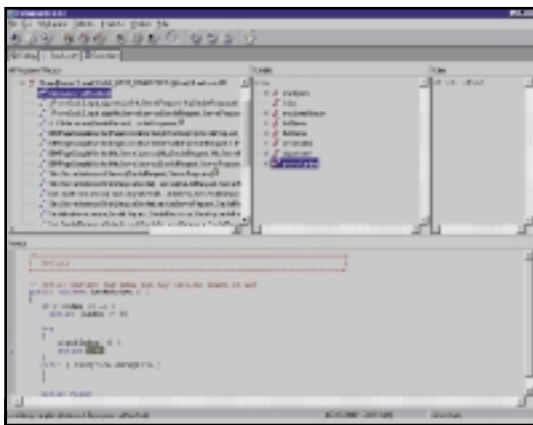


FIGURE 4 Interactive debugging

Launch WebSphere Test Environment” from the IDE Workbench window. This starts the built-in WTE Web server and the associated servlet engine. Behind the scenes, this menu item runs the com.ibm.servlet.SERunner class with predefined properties. To test JSP code and servlets, invoke your favorite Web browser with the appropriate URL. For our sample application we point our browser to <http://localhost:8080/PhoneBook/PhoneBookInput.html>, which displays our search criteria input page. After we’ve pressed the Search button to submit the request, our PhoneBookServlet is invoked. Finally, our PhoneBookOutput.jsp is called to present the search results. Figure 2 illustrates the flow of program execution.

JSP code is supported by the WTE, which compiles a JSP file into a servlet and runs it. Compilation occurs the first time the JSP file is requested and whenever the JSP file changes. A JSP Execution Monitor allows you to step through JSP code and debug its corresponding generated servlet code, which is automatically imported into the IDE. Given this functionality, you can seamlessly trace and debug the JSP code and all the scriptlets and JavaBeans used by it. To configure the JSP Execution Monitor, invoke the cascading menu item “Workspace | Tools | JSP Execution Monitor...”, which displays a dialog box with the various options available. Once you’ve enabled the IDE to monitor JSP code execution, the JSP Execution Monitor window will be launched whenever a JSP file is called. As Figure 3 shows, you can trace the execution of the JSP file at source-code level, and you can watch the corresponding generated servlet Java code as well.

To debug the generated servlet code, you need to set breakpoints in the servlet, which has been automatically imported into the JSP Page Compile Generated Code project inside the IDE after successful JSP page compilation. From there you simply run and debug the servlet as you would for any other Java code in the IDE.

Incremental Compilation and Hot Linking

In other programming environments, after you’ve made code changes to your servlet, JSP code or JavaBeans used by the JSP file, you have to stop and restart the Web server to update and reload the corresponding Java classes. During your daily RAD, source modification is a typical, routine coding activity. Having to stop and restart the

Web server and plug-ins can be frustrating, and can significantly slow down the edit/test/debug development cycle.

VisualAge for Java supports incremental compilation and hot-linking. When you make changes to a method in a Java class, only that method gets recompiled. In the IDE you can modify code in the debugger and continue to run with the new code without exiting or restarting the debugger. You can reset to a specific stack frame and resume running from there without having to reload the entire Java class and start from the beginning. In addition, you can run code that has errors in other parts of your Java class. In addition to hot linking and debugging interactively, you can start testing and fixing up “placeholder” code on the fly, even if the entire Java class isn’t syntactically correct and hasn’t been compiled completely in advance.

Figure 4 shows the debugger that’s built into the VisualAge for Java IDE. We use the debugger to step into the Employees JavaBean, which is used by the JSP file that we’re tracing in the JSP Execution Monitor. In the debugger window you can view the program execution stack, inspect the value of variables, modify the code and save the changes in the source view pane, and so on.

These features in the IDE provide a great boost to your productivity. Changes to your JSP code can be reloaded into the WTE, which automatically retranslates the new JSP code into a servlet. After the page compilation, you can resume debugging and unit testing with the modified code without stopping and restarting the Web browser or the WTE Web server and servlet engine. This shortens the edit/test/debug cycle and enables you to discover and debug subtle programming errors, such as bugs that would occur only after your servlets and JSP files have been running for a long time.

Conclusion

In developing server-side Web applications, it’s imperative that you be able to debug and unit-test servlets and JSP files both individually and together. The incremental compilation and hot-linking features in VisualAge for Java, combined with the built-in unit test environment and JSP Execution Monitor, increase your programming productivity and make it easier for you to interactively debug and iteratively unit-test your server-side Web applications. ☛

jinli@ca.ibm.com

AUTHOR BIO
Jin Li, a software designer at the IBM Toronto Laboratory, is currently working on VisualAge for Java and WebSphere solutions. He holds an MS in computer science from the University of Toronto and is a member of ACM and the Toronto Java User Group.

Environment (WTE) in the IDE so you can easily debug and unit-test servlets and JSP codes. The IDE includes an interactive debugger and supports incremental compilation. Servlets are tested using a Web browser that sends a request to the servlets executing in the IDE. You can set breakpoints in the servlet, debug it and make code modifications that are dynamically linked to the running code.

To launch the WTE, invoke the cascading menu item “Workspace | Tools |

Amazon.com

www.amazon.com/java

Java 2 Platform Standard Edition 1.3/ Enterprise Edition 1.2.1

by Sun Microsystems

REVIEWED BY JIM MILBERY



AUTHOR BIO

Jim Milbery is a software consultant with Kuromaku Partners LLC (www.kuromaku.com), based in Easton, Pennsylvania. He has over 15 years of experience in application development and relational databases.

jmilbery@kuromaku.com



Java 2 Platform Standard Edition 1.3/Enterprise Edition 1.2.1
Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900

Web: www.java.sun.com
Phone: 650 960-1300

Test Environment:
Client/Server: Dell Precision 410, 128MB RAM, 15GB disk
drive, Windows NT 4.0 (Service Pack 4)
Pricing: Free downloads: java.sun.com/j2se
java.sun.com/j2ee

I'll be the first to admit I was a little reluctant when **JDJ** approached me about reviewing the newest release of Sun's Java 2 Standard Edition platform. It's a little like asking a Jim Morrison fanatic to review the *Doors Greatest Hits* CD. Since I spend most of my days working and thinking in Java, I find it a little hard to be truly objective about it. (And it's even more difficult to choose which topics are worthy of inclusion in a review!) Nevertheless, the Java 2 platform is jam-packed with critical technologies for building enterprise-class applications. Sun has released new versions of both the Standard and Enterprise editions, and in this article I'll be taking a brief look at them.

Java 2 Platform, Standard Edition 1.3

Over the past 12 months much of the focus has been on the use of Java as a server-side technology. Many application developers have shifted their focus toward using Java as a technology for generating XML and HTML applications to the browser. Although this trend has certainly garnered a lot of press, armies of developers are still using Java to build client applications as well. The focus of the Java 2 Platform, Standard Edition 1.3 release is heavily weighted toward performance gains and improved deployment options for client-side applets and applications.

This is an important area of improvement for Java. Most of the enterprise hardware and software vendors have been migrating to Java as a platform over the past several years. Enterprise vendors have historically supported multiple platforms with their applications. There were many inconsistencies between applications on these different platforms because vendors were forced to code at the operating-system level. The version of any given program on Windows was often very different from its counterpart on UNIX in terms of both functionality and user interface. The advent of common "platforms" of functionality such as OSF/Motif solved part of the problem, but inconsistencies between platforms were still the order of the day. The Java platform has been a unifying umbrella for these enterprise vendors. Java provides a consistent set of interfaces to the operating system and a common user interface layer across multiple platforms.

Consider the case of Oracle Corporation. Oracle8i comes equipped with a database management utility in the form of the DBA Studio. Since it's written in Java, database administrators have a standard, consistent interface with which to manage databases, regardless of the platform. Countless enterprise vendors have adopted this technique of

using Java Applets and Applications as the standard deployment platform for what I call *rich-client* applications.

Technology enthusiasts tend to paint things with broad brushstrokes. When Java first appeared, we jumped all over ourselves to build client applications using applets. Then, when we realized there were some potential issues with this approach, we went in the opposite direction – lock, stock and barrel. The truth is, we need client-side Java *and* server-side Java. While Java Servlets may be ideal for developing dynamic HTML applications, they don't offer a rich enough user interface to deliver an application such as Oracle's DBA Studio. As usual, we were both right and wrong about client-side Java.

The official Reviewer's Guide for the Java 2 Platform, Standard Edition v1.3 release highlights five different classes of improvement (four of which apply to rich-client development):

- Performance
- Web deployment
- Interoperability
- Security
- Development productivity

PERFORMANCE

Sun promotes the J2SE 1.3 release as the fastest release of the Java platform thus far. One of the major complaints with rich-client Java applications has always been performance. Operating system and user-interface independence are important goals, but inferior performance is a heavy price to pay for such portability. Sun has included a new production version of their HotSpot technology into J2SE that improves the performance of rich-client applications. They've also made strides in optimizing and tuning the J2SE class libraries for faster execution, improved application start-up and reduced resource (memory and disk) usage. While Sun makes official performance claims regarding HotSpot, the best resource for more information on performance is the Java Developer Connection forum on their Web site. The general impression I got from scanning the forum entries was that developers have found HotSpot extremely beneficial for improving performance – especially for JDBC-based applications. While HotSpot was an optional add-on for previous releases of the SDK, it's the default JVM for 1.3. The primary performance improvement technique within HotSpot is the dynamic analysis and compilation of critical portions of your code – thus the name *HotSpot*.

WEB DEPLOYMENT

Sun has also made some improvements in the area of Web deployment for when you need to deploy rich-client applications using applets. J2SE 1.3 supports the concept of applet caching. Frequently used applets can be cached locally on the desktop (without violating any of the security mechanisms). The applet cache is

Quest

www.quest.com

maintained separately from the browser cache with the J2SE, which ensures that applets won't get flushed from the client cache unless the applet itself is changed on the server. Applets can now have optional packages included as part of their definition, and these additional APIs can be downloaded (and installed) as necessary when the host applet is run.

INTEROPERABILITY

J2SE now sports a production version of the CORBA IDL compiler along with support for RMI (Remote Method Invocation) over the IIOP (Internet InterORB Protocol). This is an important consideration for linking Java applications to preexisting code that may have been written in other languages, such as C or COBOL. In this brave new world of application interoperability you may find the need to link your newfangled Java applications back into legacy code. While vendors have offered their own extensions for supporting CORBA, the J2SE 1.3 release makes this a standardized part of the Java language.

SECURITY

Security has been enhanced in several critical areas. J2SE 1.3 supports JNDI (Java Naming and Directory API), which allows developers to use directory servers with Java applications. This is an important consideration in making Java part of the overall infrastructure of a corporate application. Directory servers and JNDI offer you the ability to implement single sign-on services for applications and consolidated access to resources such as print and file servers. J2SE 1.3 also provides new support for RSA signatures and dynamic trust management, and by including standard RSA signatures from VeriSign and Thawte, developers can include RSA security without having to write their own service provider. Equally important is the fact that J2SE 1.3 provides dynamic pop-up dialogs that allow users to validate applet signers on an as-needed basis. This eliminates the need to preconfigure each client machine with a cache of trusted certificates for every applet that could potentially be accessed. This feature will make it a lot easier for software vendors to release versions of their software as applets as well as applications. In particular, certain types of management applications (database-monitoring tools come to mind) can be deployed as applets, allowing administrators to access these interfaces from remote workstations.

DEBUGGING

One of the most significant improvements in the J2SE is the addition of the Java Platform Debugger Architecture (JPDA). Ideally, developers should be able to use the best-of-breed approach when it comes to selecting Java development tools,

application servers and database servers. However, two critical issues have forced developers to consider integrated "stacks" of technology over best-of-breed multivendor solutions. The first issue is the evolution of the EJB specification (addressed by J2EE) and the second is the difficulty in debugging multitiered applications. With the JPDA it becomes possible to debug the various tiers of your application (client, application server and database) within the familiar confines of your favorite Java IDE. The JPDA architecture is composed of three layers as shown in Figure 1.

The first layer is the JVMDI, which defines the debugging services that a virtual machine must provide. Vendors that include a custom JVM within their products (such as database engines) can expose the VM to developers through the debugging API. Underneath the covers is the JDWP, which provides the format for information and messages that must be transferred between the debugger interface and the physical process being debugged. The last layer of the JPDA is the Java Debug Interface (JDI). Java IDE developers can use the JDI as the language interface for constructing remote debuggers. The inclusion of JPDA within the J2SE 1.3 release offers software vendors the ability to provide for higher levels of interoperability between the various layers of the application deployment topology. Most of the top technology vendors have already endorsed the JPDA architecture and have pledged support for it within their product lines.

Java 2 Platform, Enterprise Edition 1.2.1

While the two editions of the Java 2 platform are tightly intertwined, the focus of the Enterprise Edition is aimed squarely at the middle and database tiers of the application architecture. The J2EE model has been designed and architected to alleviate the problems associated with two-tier applications. The core of the J2EE model is the ability to divide the application work into two parts: business/presentation logic and standard system services. Application developers are responsible for constructing the business rules and presentation logic, while the multitier system services are provided by the J2EE platform.

Ten key technologies are defined as part of the Java 2 Enterprise Edition specification:

- Enterprise JavaBeans
- JavaServer Pages
- Servlets
- Java Message Service
- Transactions
- J2EE Connector
- CORBA
- JDBC
- XML
- JNDI

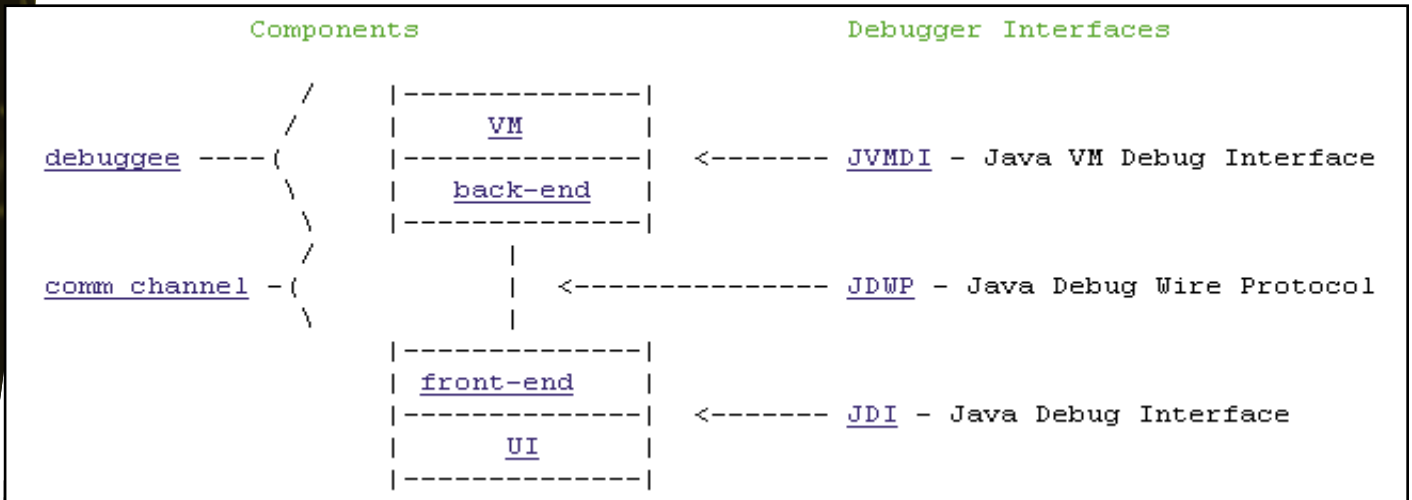


FIGURE 1 Java Platform Debugger Architecture

Zucotto

www.zucotto.com

It's difficult to argue which of these technologies are the most important, but most technologists would agree that EJBs (Enterprise JavaBeans) are at the top of the list. They're at the core of the J2EE platform. While Sun released a new version of the EJB 2.0 specification at the JavaOne conference, the current version of the J2EE platform includes support for the EJB 1.1 specification. The jump from EJB 1.0 to EJB 1.1 has been an important hurdle for application server vendors. Previously, vendors were required to provide support only for session beans, and the implementation of the descriptors for EJBs was somewhat ambiguous. EJB 1.1 requires support for entity beans (persistent objects) and makes use of XML descriptors. Ultimately this is a good thing for developers. The tighter the specification, the more likely it is that vendors will create consistent and compatible EJB products. In the longer term this means that developers should be able to create EJB components using one set of tools and applications – and be able to deploy these objects to servers provided by other vendors. Up until this point the various vendors that have implemented EJBs have tended to add proprietary extensions (particularly in the area of deployment). We aren't there yet, but EJB 1.1 is a step in the right direction, and EJB 2.0 goes even farther.

JavaServer Pages is an extension of the Java Servlet API, and it's one of the most exciting aspects to the J2EE platform (at least in this reviewer's mind). Java Servlets were introduced as a technique for providing CGI-style scripting using the Java language. JSP is a natural extension to Servlets – one that's easier to author and more compatible with popular HTML design tools. JSP provides XML-style tags and scriptlets that allow developers to build dynamic Web applications and keep the presentation logic separate from the content. JavaServer Pages and Java Servlets provide the front end for browser-based applications in the J2EE architecture. They can be used on their own and in concert with EJBs to create enterprise class applications. While Java language programming can be complex for novice developers, JSPs and Servlets are much less daunting. Furthermore, the design of JSPs lends itself to the creation of prebuilt "tags" (just like XML tags) that can be distributed as libraries. This allows developers to create best-of-breed tags that can easily be used across applications. Although Sun owns the JSP and Java Servlets specification, they have turned over the reference implementation to Apache under a project known as *Tomcat*. By doing so, they're making world-class JSP/Servlet technology available to the masses. Even if your organization isn't ready to work with some of the other (more advanced) components of the J2EE, you'll be able to leverage the power of JSPs and Servlets.

Listing 1 shows the host JSP which in turn uses the "include" directive to call the HTML code shown in Listing 2.

Executing this simple JSP produces the HTML page shown in Figure 2.

The simple elegance of JSPs allows the J2EE to involve more customers and provide a large community of developers with the chance to work with Java in some fashion. The platform independence of JSPs and Servlets ensures developers that they'll have a wide variety of platform choices on which to run their applications. If you've been sitting on the fence when it comes to Java, now's the time to dive in – and JSPs are an ideal first step.

Conclusion

Sun continues to try to strike a balance between innovation and stability with the Java 2 platform. The J2SE and J2EE releases consolidate some of the individual APIs that have been defined along the way – and the resulting platform is packed with innovation. In addition to the components discussed in this article, the J2EE also contains asynchronous messaging with JMS and transaction management

with JTS/JTA. Although the J2SE/J2EE specification is somewhat rigorous, there's still plenty of room for interpretation and this is where the real fun begins. All of the J2EE licensees have added their own interesting extensions (where appropriate). So, despite the fact that they're all working off a single standard, each individual product is loaded with unique customizations. I'd recommend that you read through the materials on the Java 2 platform site and take a look at the developer forums before you start talking to vendors about their products. ☘

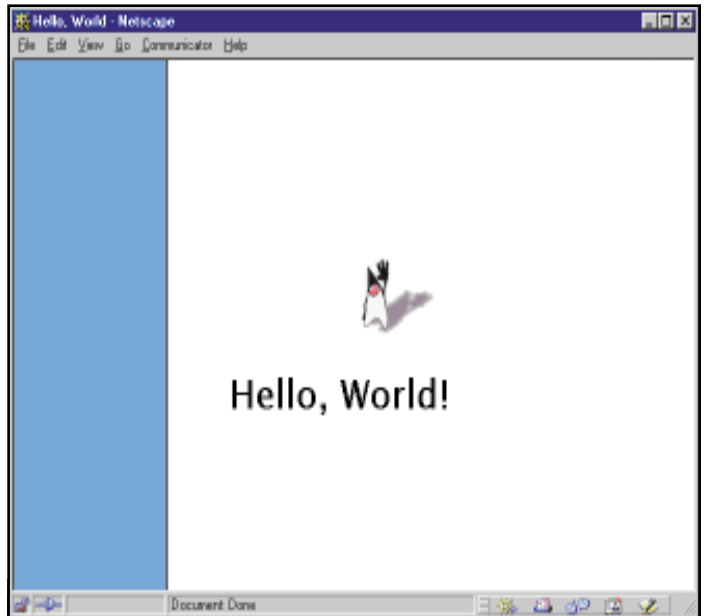


FIGURE 2 Duke's Hello, World! page

Listing 1

helloworld.jsp

```
<%@ page info="a hello world example" %>

<html>
<head><title>Hello, World</title></head>
<body bgcolor="#ffffff" background="background.gif">
<% include file="dukebanner.html" %>
<table>
<tr>
<td width=150> &nbsp;&nbsp;&nbsp; </td>
<td width=250 align=right> <h1>Hello, World!</h1> </td>
</tr>
</table>
</body>
</html>
```

Listing 2

dukebanner.html

```
<table border="0" width="400" cellspacing="0" cellpadding="0">
<tr>
<td height="150" width="150"> &nbsp;&nbsp;&nbsp; </td>
<td width="250"> &nbsp;&nbsp;&nbsp; </td>
</tr>
<tr>
<td width="150"> &nbsp;&nbsp;&nbsp; </td>
<td align="right" width="250">
 </td>
</tr>
</table>
<br>
```



Slangsoft

www.slangsoft.com

Component Assembly Using a Visual Development Environment

Accelerate the development process from concept to working Web application

WRITTEN BY
JEFF CHAPMAN AND
GERARDO DADA



very day more organizations are embracing CBD (component-based development) for its promise of code reusability and, as a result, reduced development effort and faster time-to-market. In addition, Java is quickly being adopted as a standard for Web development.

Application server vendors are now selling components for e-commerce and vertical applications. For example, BEA Systems identified this trend early on, and understood how it could become a competitive advantage. To that end, BEA recently acquired The Theory Center, a software development firm focused on creating EJB (Enterprise JavaBeans) component solutions for e-commerce. Now BEA offers The Theory Center's e-commerce objects bundled with their own application server and is marketing the solution as its Commerce Server.

According to the GartnerGroup, by 2003 at least 80% of all new applications will be built using primarily components. New vendors, such as Evergreen, are joining the component market. Component marketplaces such as flashline.com and componentsource.com are making thousands of components readily available for developers.

CBD also makes it easier to create applications that separate both presentation and business logic. The complex logic is created in an EJB while the user interface is developed using JavaServer Pages (JSP) and/or servlets.

However, CBD introduces new challenges that developers must face, along with existing challenges that are inherent to developing complex applications. Creating a harness to test an object is a time-consuming task. Assembling components is still complex, requiring highly experienced Java developers to spend

a lot of time coding. Integrating COM and EJB objects in the same application is nearly impossible.

Separating presentation logic from business logic on the user-interface side is a challenge developers face when trying to create an easy-to-maintain application. In addressing this issue of logic separation, Michael Lacy wrote in the May 2000 issue of *DDJ* (Vol. 5, issue 5): "...the merging of Java and HTML frequently creates development headaches...Once the application is finished, our headaches turn into migraines upon the realization that we have to dig through all of the intermingled code for maintenance and enhancements."

Another factor is finding, training and retaining experienced Java developers. Even with thousands of developers learning Java and other advanced skills, the supply of qualified candidates is unlikely to catch up with demand anytime soon.

As development organizations struggle with these challenges, there's a growing demand for development tools that hide the complexity of an application and enhance productivity. A development environment is available that enables developers to visually assemble components while hiding their logic; this makes teams more productive by separating tasks according to the developer's skill level. This environment, Tango 2000 from Pervasive Software, is centered around visual programming

techniques that accelerate the development process and enable developers to work at different levels based on their skill sets.

Tango 2000 makes it easy to build Web applications that incorporate a variety of reusable components. Currently it allows users to easily mix COM and JavaBean objects to implement business logic. A forthcoming Tango 2000 service pack will enable developers to incorporate Enterprise JavaBeans into Tango 2000-based applications.

Tango 2000 also introduces a new construct called a Tango Class File (TCF) that makes it simple to encapsulate frequently used logic into reusable components and to separate business logic from presentation logic.

Developers will be able to visually integrate COM objects, JavaBeans, TCFs and EJBs in the same manner. Once the object is located, Tango introspects it and finds its methods, dependent objects and parameters. Passing information to and from any of these object types is also possible within the Tango environment using Tango-specific logic. For example, depending on the developer's skill and the complexity of the function, a database query can be performed through a wizard, either visually using a Query action or with a direct DBMS command writing specific SQL code. Sending an e-mail is another function that can be done quickly and visually using Tango actions.

Sybase

www.sybase.com/products/server

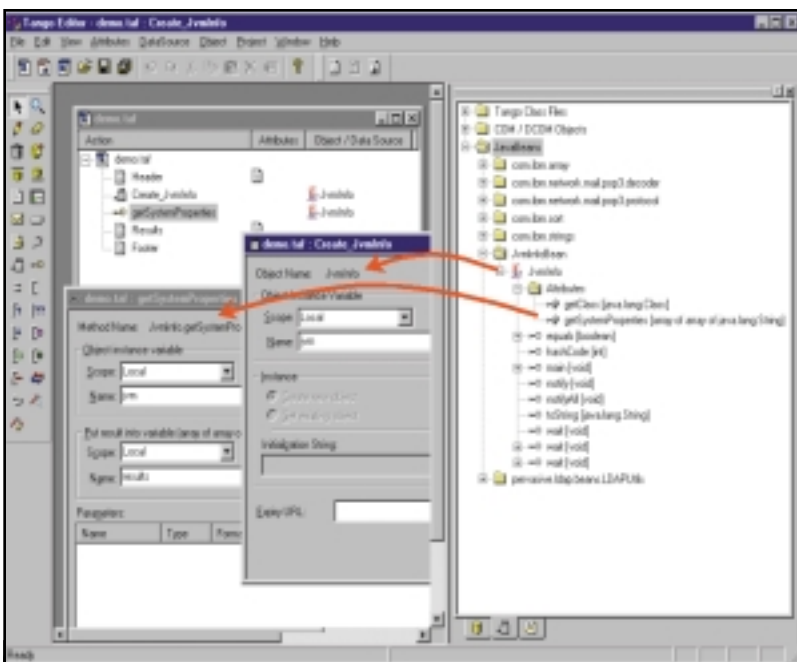


FIGURE 1 Using JavaBeans with drag-and-drop simplicity



FIGURE 2 LDAP search sample

Using a Simple JavaBean Visually

Using a JavaBean or COM object is a three-step process:

1. Navigate to the object and add it to the Tango Object Workspace. Tango 2000 introspects the object to enumerate any dependent objects. When you select the desired objects from the list, they're introspected further to enumerate the public methods and properties, as well as their parameter data types.
2. Create an instance of the object by simply dragging it into your project. Give it a unique variable name and assign a scope to it. Tango provides a variety of scopes ranging from local through user to application scope. The life cycle of the object is governed by the reference count for the object's scope. When the object goes out of scope and there are no other users, the object is released.
3. Use the object's methods and attributes by simply dragging them into your project (see Figure 1). Specify where to put the results, usually a local variable. Then for each parameter you can optionally provide literal values or Tango variable names that get resolved at runtime.

Integrating Advanced Services

By leveraging COM and JavaBean objects, you can integrate a wide variety of technologies into your Web application. Tango 2000 will automatically convert, by default, the results of object calls into HTML. Returned Java arrays are automatically converted into HTML tables that inherit the default style settings for tables. Encapsulating style settings in TCF methods makes it easy to adapt the Web site's themes with a single change.

It's important to use objects that are specifically designed for server-side usage, since they have no dependencies on a developer to operate a GUI. There are plenty of available COM objects suitable for server-side usage. For example, Microsoft offers the Windows Scripting Host object, the FileSystem object, and the Microsoft XML and XSL advanced parsers.

There are numerous sources of JavaBeans as well. For example, IBM's alpha-Works Web site has an interesting collection of JavaBeans. You can also use the JavaBeans that come with a variety of SDKs. Novell and Netscape also provide JavaBeans with an interface to their directory services.

Making Object Reuse Even Easier

Some objects are specifically designed to be used only in pure Java applications.

Intuitive

www.optimizeit.com

Occasionally you may prefer to wrap these JavaBeans in your own to simplify the reuse of the technology. The methods in the Netscape Directory Server SDK are good examples. In this case five LDAP operations are associated with doing a simple LDAP search (see Figure 2). Instead of doing all these tasks directly from Tango, you might craft a JavaBean with a method to encapsulate these steps. You also might want the JavaBean to return the LDAP results as an XML document. To simplify reuse of this JavaBean, the developer can create a collection of snippets that provides many simple and advanced examples of method invocations.

You could just as easily implement a COM object to interface with the ADSI (Active Directory Server Interface) on Windows platforms. In fact, if you build a COM wrapper for ADSI with the same interface as the JavaBean wrapper for the LDAP server, the same Tango 2000-based application can be constructed to use either directory service.

Enterprise JavaBeans Made Easy

With the next Tango 2000 service pack (planned for this summer) you'll be able to use EJBs in your Tango applications. As before, simply navigate the Tango Editor to the location of the JAR, WAR or EAR to add it to your Object Workspace. You'll also be able to view the methods and properties of the EJB as easily as those of COM, JavaBean and TCF objects.

With the release of this new version, you won't have to deal with the complexities of Home and Remote interfaces, Deployment Descriptors, Application Server variations and so forth. Instead, you'll be free to devote all your energies to the task of implementing your business logic and crafting suitable presentation logic.

A True RAD Approach to Web Application Development

If you currently use components in your development projects, you already appreciate how important they can be in assembling a cutting-edge Web site in the shortest time frame possible and maximizing code reuse. The typical developer in Tango 2000 is able to implement a project in a fraction of the time needed in other Web development environments because it lets you stay focused on your business logic, not on

the "plumbing" infrastructure that enables the application.

An ideal Web development team will consist of a group of individuals, each with different skill sets (see Figure 3). Senior developers needn't spend valuable time creating presentation logic or user interfaces. Instead, they can focus on the complex business logic that fully requires their experience. They can use a visual assembly tool to create a harness to test and demo the objects they're developing, and they can also create business logic inside this RAD environment.

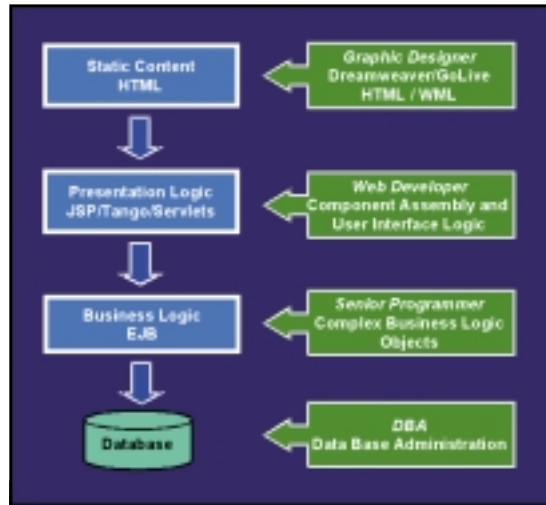


FIGURE 3 Web development team for CBD

A graphic designer typically focuses only on the presentation logic of the application using any HTML editor. The tool of choice is Dreamweaver given its easy integration into other environments including Tango 2000.

A third segment of the development team – Web developers with intermediate development skills – can be very productive assembling finished objects created by senior programmers, integrating them with the presentation logic and creating business logic that doesn't require the complexity of Java. These tasks can be accomplished quickly and easily with the Tango environment.

The catalog and shopping basket pieces of an e-commerce application typify business logic that can be developed entirely within the Tango environment, usually in less time than Java. The back-end integration with systems such as accounting and shipping is typically handled with an object. Tango provides native support for XML documents, allowing you to build portable and scalable B2B or B2C applications. Wireless applications are also created in a snap, given Tango's built-in WML support.

Efficient Code Reuse

Tango 2000 provides an extensive collection of snippets to simplify tasks ranging from Tango 2000 server configuration tuning to WML page crafting. Team members can create and share libraries of snippets, thus simplifying their reuse by hiding the complexity of object interfaces and avoiding typographical errors.

Pervasive Software offers its own collection of TCFs and other objects to simplify development efforts. Developed by customers as well as by Pervasive employees, these objects are available in the ComponentZone on the Pervasive Web site (www.pervasive.com).

Summary

When developing at Internet speed, you need an application server and a development studio that delivers on the promise of rapid Web application development. Tango 2000 allows developers to accelerate the development process from concept to working Web application in days rather than weeks or even months.

The Tango 2000 Development Studio is available for Mac OS and Windows 95/98/NT/2000, while the Tango 2000 Application Server is available on Windows NT, Linux, Solaris and Mac OS. Applications developed in Tango are truly cross-platform, so as your business grows, you can move your applications to more scalable platforms. Tango 2000 also supports load balancing across clusters of servers, allowing you to easily scale your Tango application to match your requirements. Tango 2000 Application Server also supports caching and database connection pooling to provide optimal performance for your Web applications.

Pervasive Software plans to release a new product before the end of the year that will allow developers to deploy Tango-developed applications to J2EE servers. This product will allow developers to create EJBs and servlets, as well as integrate objects using the same visual approach used to create Web applications in Tango.

CBD promises a more efficient way to develop applications with commercial, vendor-provided or internally developed objects. Using a visual development environment to test and assemble these objects significantly enhances productivity for all the members of the development team. ☺

CBD promises a more efficient way to develop applications with commercial, vendor-provided or internally developed objects. Using a visual development environment to test and assemble these objects significantly enhances productivity for all the members of the development team. ☺

jeff.chapman@pervasive.com

gdada@pervasive.com

AUTHOR BIOS

Jeff Chapman is a senior lead quality engineer at Pervasive Software.

Gerardo Dada is a Tango product marketing manager at Pervasive Software.

Flashline

www.flashline.com

Building Out the Internet Platform: A History of Application Servers

From a simple collection of code libraries to a robust, standardized platform

WRITTEN BY
PHIL COSTA



It wasn't long ago that many developers didn't know what an application server did. These days it's become part of our common vocabulary. The main reason for this shift has been the rapid growth in the importance of the Internet as a platform for business applications. Without application servers the Internet would be a much less exciting place. This article shows how these vital pieces of the Internet infrastructure have evolved and explores where they're headed.

Why We Have Application Servers

Viable product categories emerge for a very simple reason: they fulfill a compelling need for a large group of users. In the case of application servers, developers needed a product that could assemble all the low-level resources used in Internet applications and encapsulate

them in a productive framework that would enable the construction of rich and interactive Internet applications. In this sense application servers closely resemble desktop operating systems, which continue to enable an entire industry of productivity tools for business users and consumers alike. However, whereas an operating system encapsulates

a particular set of hardware devices and windowing services, application servers encapsulate the protocols and data types used on the Internet as well as a different set of user interface services.

As with all infrastructure products, application servers have had to evolve with the needs of developers. Just as operating systems have changed dramatically in the past decade, the application servers of today are but a shadow of the Internet business platforms that will emerge in the next few years. To better illustrate where application servers are headed, it's useful to review the steps that brought us here.

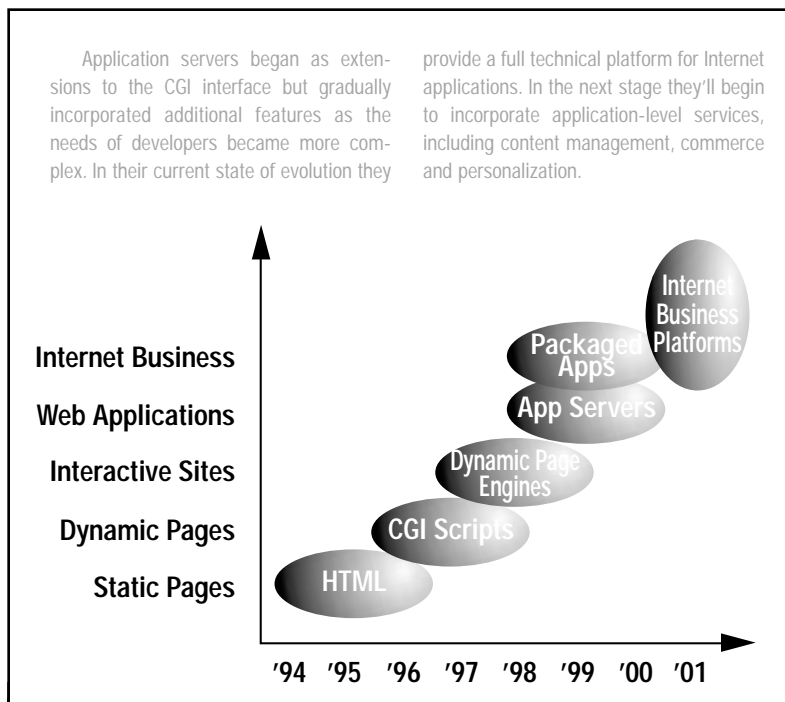


FIGURE 1 The evolution of the application server

History of Application Servers

The history of development on the Internet is relatively short, but the challenges have grown by several orders of magnitude in that period. As illustrated in Figure 1, the goals of Web developers have evolved from adding dynamic data to previously static pages and building interactive sites to the full-fledged applications deployed by Amazon.com or your online bank. At each of these stages the requirements of what became the application server have expanded to include new resources and functionality. We'll revisit the stages to come a little later on; first we'll take a look at the products

ThinWeb

www.thinweb.com

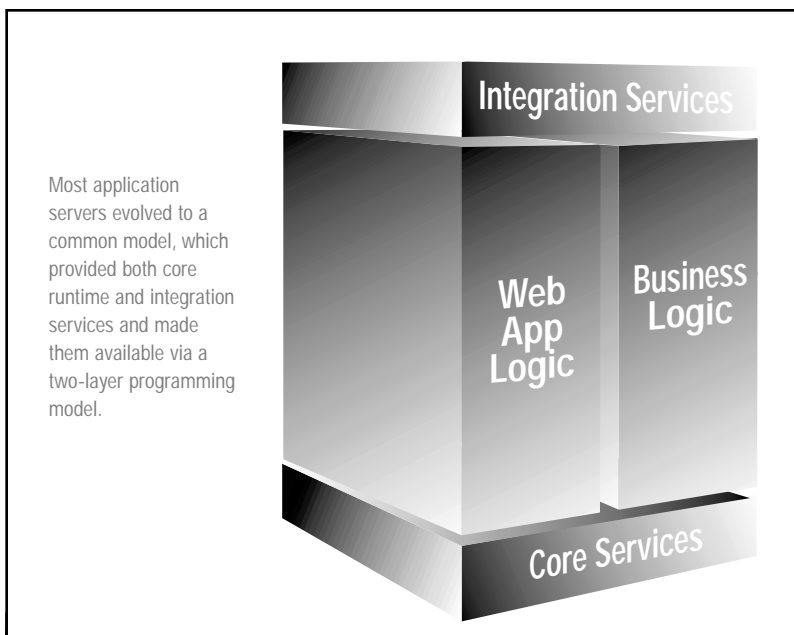


FIGURE 2 Application server architecture

and technologies that formed the genesis of the application server.

The technology that first enabled the transition from static to dynamic pages was CGI (Common Gateway Interface). Unfortunately, using it left plenty of work for developers. They not only had to write the business logic for their page, but each CGI program had to implement its own solution for simple tasks such as opening connections to a database, formatting the results for delivery in HTML and maintaining session information between requests. As more users tackled the same problems, they began to circulate libraries of reusable code that solved some of them. However, as projects became more ambitious, assembling these point pieces into a complete application that worked reliably became an increasingly difficult task.

At this point some of the more entrepreneurial types recognized an opportunity, and the Internet application server market was born. These pioneers wrote their own set of reusable resources (e.g., database connections, state management, results formatting, pooled CGI connections) and packaged them so developers could be more productive than they had been with ad hoc code libraries.

Initially, most of these products were sold as tools for Web developers. Like RAD tools before them, they made their runtime resources available through a scripting language and/or a set of objects that encapsulated the underlying plumbing. Some of the earliest examples include Allaire ColdFusion, HAHT Software HAHTsite and Netscape

LiveWire. They're pictured collectively in Figure 1 as dynamic page engines.

Meanwhile, people and businesses were flooding onto the Web and recognizing the incredible opportunities offered by applications deployed on it. As a result, the more aggressive developers quickly moved beyond simply generating dynamic pages. To ease this transition, they wanted tools that could help them build full-fledged applications with much more complicated requirements.

To respond to these demands, software companies began to expand the range of plumbing they provided. To increase scalability and performance, they incorporated design principles and/or technology from TP (transaction processing) monitors. To simplify integration with existing corporate systems, they began to build adapters for e-mail systems and directories and to integrate ORBs (object middleware) for linking to legacy applications. As the applications being built became more important to the business, they also began to incorporate security tools such as authentication and encryption.

As these pieces were being added, server providers continued to enhance the tools and programming languages that provided access to those resources with the goal of making developers even more productive. Thus the scripting languages grew in power and the collections of objects became systems in and of themselves.

As the runtime services provided by these products gradually became richer, it became obvious that this new breed of product was much more than a tool, but

rather a new piece of infrastructure required by anyone building significant applications for the Internet. It took an additional step toward standardization, however, before the term *application server* became widely accepted.

Each of the early application servers took its own path to maturity, but by the middle of 1999 most began to converge around a similar architecture (see Figure 2). The plumbing consisted of a set of core runtime services (e.g., load balancing, transactions) as well as integration services that offered connectivity into other systems (e.g., databases, file servers, e-mail servers, applications). To access these resources, developers generally had a two-layer development model. This model consisted of a Web application logic layer that handled presentation of data to users and a business logic layer that handled data access and data processing. Once this model was widely accepted, the name *application server* became more widely known.

Emergence of Standards

In the past 18 months the application server arena has seen even greater change as application server providers have moved to standardize the architectural model and the interfaces through which developers access the server's resources.

Initially this standardization came in pieces. For instance, Java Servlets and later JavaServer Pages caught the interest of many Web developers as a powerful way to use Java for Web application logic (i.e., generating dynamic Web pages). Microsoft's ASP (Active Server Pages) and Allaire's CFML (ColdFusion Markup Language) have gained widespread support at this layer as well. Similarly, EJB (Enterprise JavaBeans) and Microsoft's COM/COM+ have emerged as standard models for developing business logic components.

More recently, Sun and its partners in the Java community have tried to rationalize these individual pieces into a uniform and complete collection of APIs, now termed the Java 2 Platform, Enterprise Edition (J2EE). The APIs in J2EE provide a comprehensive set of services and a two-layered programming model for developers building applications on the Internet, with Servlets/JSP at the Web application logic layer and EJB at the business logic layer. The collection of APIs known as J2EE now has the support of most of the major players in the Internet software market, including Allaire, Art Technology Group, BEA, Bluestone Software, IBM, Oracle and Sybase.

New Atlanta

www.servletexec.com

The most notable exception on the list is Microsoft. In opposition to Sun's efforts to promote Java, Microsoft has developed its own set of APIs, referred to as Windows DNA (Distributed interNet Applications Architecture). There are significant differences between Windows DNA and J2EE. For instance, Windows DNA is tied to the Windows operating system whereas J2EE runs across operating systems. J2EE has been developed and implemented by a number of vendors; Windows DNA is available only from Microsoft. However, from an architectural level Windows DNA closely resembles J2EE, providing a Web application layer (ASP) and a business logic layer (COM+) as well as a set of core services and integration services (Windows 2000 and BackOffice). Moreover, given the ubiquity of Microsoft technology, Windows DNA will remain an important programming model for the foreseeable future.

While the shift toward standardized programming models (Windows DNA and J2EE) is still underway, it promises to bring enormous change to the industry and to developers. Just as the standardization of the desktop APIs around Win32 enabled a rich applications market to emerge, having a common set of programming interfaces for application servers should enable independent software houses to build packages that run on multiple application servers. While complete portability across application servers that support the same standard is still an unrealized promise, it should enable independent software vendors to dramatically reduce the cost of porting their applications from one server to another. Even without code portability, a key benefit for developers will be the ability to apply the same development skills to multiple servers and operating systems.

Unfortunately, standardization also has its drawbacks. For instance, because J2EE hides the underlying details of the operating system on which the server is running, applications built with it often don't take advantage of the rich services offered by today's server operating systems.

In addition, J2EE is a complex model for developers to swallow. Accommodating all of the needs of developers under one umbrella has meant adding features that are required only by advanced developers. As a result, one of the major challenges for application server providers will be to encapsulate the J2EE platform in a way that makes all developers productive, even those building simple applications. One promising approach taken by several

vendors is to build high-level JSP tag libraries that encapsulate Java platform services for developers accustomed to tag-based languages. Another is to take an existing language that has proved itself in the market and integrate it much more closely so it can take advantage of Java platform services while affording developers the same productivity they've enjoyed in the past.

The other major challenge is one application servers have faced since their origin: the need to incorporate additional technologies as they become part of the developer arsenal. For instance, as portable devices become increasingly connected, their users want access to the same business logic that's now available on the Web. However, since these devices have user interfaces and bandwidth restrictions dramatically different from those of a browser, they require a different set of technologies at the Web application logic layer. Thus we're already seeing a flurry of activity as vendors incorporate technologies like WML (Wireless Markup Language) to support these new types of applications within the application server.

Along the same lines, many companies are now looking to connect applications built using application servers directly to other application servers, not to new users. Thus the product definition is evolving to include new technologies for application integration and B2B messaging. Two promising technologies in this area have received wide support: the Java Message Service (JMS), a Java-based API for asynchronous messaging, and XML-SOAP (Simple Object Access Protocol), an XML-based protocol for invoking applications using HTTP.

Future of Application Servers: Internet Business Platforms

As this short history illustrates, application servers have been successful because they evolved with the changing needs of developers, incorporating new technologies (e.g., XML) and supporting new types of applications as they emerged. However, even as application servers continue to change, the market is already entering its next phase – the expansion of the application server into a full Internet business platform (see Figure 1).

Just as developers learned that many of the most common tasks involved in Web development could be packaged in a reusable fashion, companies building their businesses on the Web are recognizing that many of the most common applications can be bought and cus-

tomized instead of built from scratch. The next few years will see application server vendors building suites of applications that meet the most common needs of businesses.

Obviously, many applications will be built by independent solution providers, and many others will be custom built by companies that use them. However, just as all businesses need systems for managing finances and human resources (the foundations of the ERP market), we're now seeing that businesses using the Internet for strategic advantage will need content management, commerce software and personalization to be successful.

Content management applications will enable developers and business users to coordinate the creation, deployment and syndication of Internet content. Commerce applications will provide the building blocks for doing business on the Internet, whether selling directly to customers or to businesses, participating in online marketplaces or managing relationships with partners and suppliers. Finally, personalization systems will underlie both of these systems, enabling companies to customize their content and applications according to the needs and roles of a particular user.

Since application servers provide the platform for development on the Internet, who better to provide these core applications than the platform providers themselves? They have the technology and expertise, and by combining an integrated set of applications with a standards-based platform they can provide business users with rapid time-to-market without taking away the flexibility and standards support valued by developers. Moreover, since the application server itself is becoming increasingly standardized, vendors need additional ground on which to differentiate themselves.

• • •

The application server market has come a long way from its humble origins. Moreover, as the Internet becomes a vital channel for global business, the application server market is poised for even greater change, evolving to incorporate new technologies such as WML and XML-SOAP as well as more application-level services. In an effort to provide developers with a productive way to develop applications on the Internet, the application server has grown from a simple collection of code libraries to a robust, standardized platform for modern business. ☛

pcosta@allaire.com

AUTHOR BIO

Phil Costa is the senior manager of strategic marketing at Allaire Corporation. His responsibilities include marketing for the Allaire Business Platform as well as research into the future directions of the Internet software market.

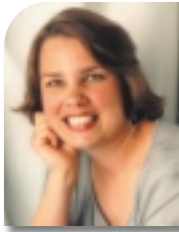
InetSoft

www.inetsoftcorp.com

Java Application Servers: Buyer Be Aware

'HTML guru wanted in Oakland to build a complex B2B commerce site'

—ad from consulting Web site, April 2000



WRITTEN BY
ANITA OSTERHAUG

Yes, Virginia, there are still people who think you can build a complex B2B e-commerce Web site using HTML. The good news is, these people are mostly harmless; they'll learn the error of their ways quickly and probably before their projects are too far down the road. There are far more insidious and dangerous misconceptions out there in the IT world regarding e-business systems.

At this point in Web history most of us know that an effective e-commerce site isn't brochureware, nor can it be built with client/server technology. Many businesses have chosen to base their Web commerce systems on J2EE (Java 2 Platform, Enterprise Edition) and its EJB (Enterprise JavaBean) components, as J2EE is intended to be a complete platform for Web-enabled, multi-tier, secure, transactional Java applications. J2EE promises better quality, maintainability and portability of systems. With the popularity of J2EE a number of vendors are offering application servers, e-commerce platforms aimed at providing object execution engines, transactional control, security and other services. These products help separate business objects from presentation logic so sites can assemble content dynamically, and content and business rules can change in real time.

No more tedious infrastructure development for us, right? We can now revel in our choices of application servers that package these and other commerce services. Well, here's the rub: all application servers aren't created equal. Despite J2EE standardization, there are still profound differences in the capabilities and scalability of these systems.

B2B Application Servers for the Discerning Designer

Most of the application servers on the market today were designed for early e-commerce Web sites that were shopping-oriented or CRM-oriented. Trans-

actions were usually simple, and often the greatest challenge was getting personalized content to a browser. Today businesses are looking to build complex B2B systems in which transactions are more precise, applications more diverse and performance more critical. Look at the requirements and characteristics of a digital exchange, a typical B2B e-business system (see Figure 1).

- The system must tie together information, such as product catalogs from many suppliers, and applications, such as order management, scheduling and auctions, in real time.
- The site must implement complex business rules to customize and secure business transactions between customers and suppliers.
- It must scale to service thousands of concurrent customers initiating com-

plex transactions with information drawn from hundreds of partners and suppliers.

- The development team can't rely on prepackaged software to provide the complex business rules, sophisticated transaction control or flexibility required by the application.
- Competitive advantage comes from nonstop performance and availability. The application can't be taken down for updates, performance tuning, system reconfiguration or system failure.

When selecting application servers for complex B2B systems, designers need to look beyond all the happy hype and ask vendors some hard questions:

- How flexible is the server's application model? Does it support more than

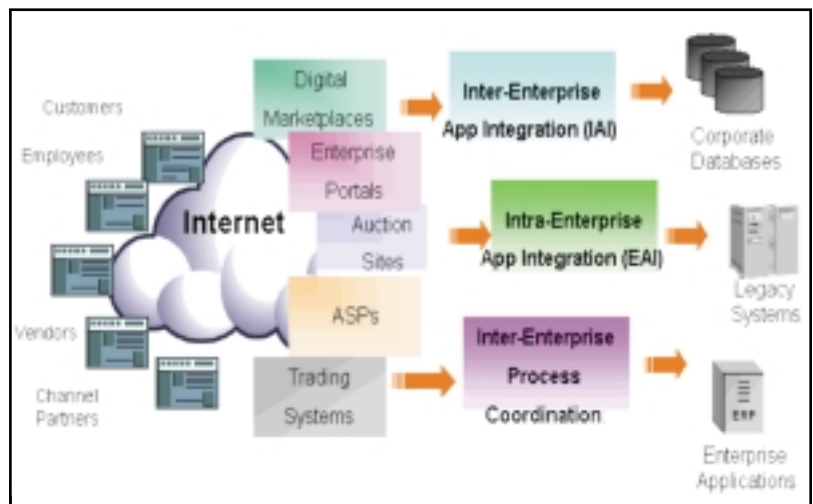


FIGURE 1 B2B e-commerce applications

Elixir

www.elixirtech.com/elixirde

just EJB applications so it can more easily integrate other internal and external business systems?

- How scalable and available is the server? Does it support the kind of nonstop, high-speed, high-volume service required by complex B2B applications without unnecessary hardware, software or administrative costs?
- How flexible are the security services offered by the product? Do they include multilevel access control that can change dynamically?
- How well integrated are all the services offered by the product? Are they patched together by the vendor through recent acquisitions (meaning you may end up doing the real integration), or was the system designed from the ground up so that services work together in a reliable, consistent way?

The answers to these questions may determine whether designers enjoy accolades or massive headaches a few months down the road.

Universal Application Models

Above all, B2B Web sites must be flexible. You need an application server that supports easy integration and deployment of diverse application models.

Consider our digital exchange company example (see Figure 2). Already this application combines several architectures into one application or set of components: EJBs, CORBA objects, JSPs and servlets. A common, well-integrated environment is crucial for success. But there are more integration challenges ahead. Today this system does daily batch loads from affiliate catalogs; when prices change from one day to the next, the company absorbs the cost differences. In the future this company's goal is to integrate with supplier systems to update prices in real time.

Integration like this calls for an application server that supports a diverse set of applications and requirements. Such a server must be designed for heterogeneous business environments:

- It must be Java-centric, not just EJB-centric. Business objects may be written in pure Java and accessed through EJBs, but they should also be accessible through CORBA wrappers, RMI (Remote Method Invocation) or other methods.
- XML parsing and support should be built in so the application server can integrate quickly with existing busi-

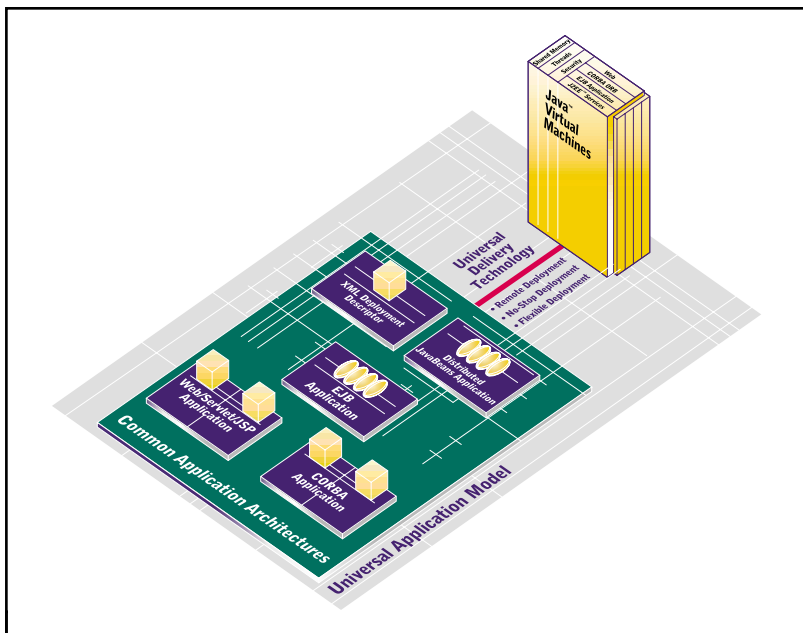


FIGURE 2 Application servers must support deployment of diverse application models.

ness systems and take advantage of standard XML APIs developed by various vendors and standards organizations.

If you employ multiple application models in your system mix, the problem of deployment becomes more complicated. Thus the application server should also include a universal deployment model that can be used to deploy EJB, CORBA and Web/servlet applications. Standardized XML-based deployment descriptors can be used for each application model, forming the basis for a common deployment process.

Comprehensive Scalability

Scalability can make or break an e-business strategy, yet this aspect of application server performance is often taken for granted in the decision process. Application server architecture makes a profound difference in application performance; problems often surface after a site is live, causing interrupted service, embarrassment and lost revenues.

For complex B2B sites we need to define scalability more broadly than in the past. To be scalable, a system must include the ability to handle peak loads in terms of users, information and transactions, and to maintain service and performance through resource failures, maintenance and upgrades.

Cost-Effective Scalability

In its quest for scalability the J2EE world has embraced clustering, the use

of shared redundant resources as a solution. But early J2EE clustering architectures have focused mainly on failover, with limited gains in scalability. Early clustering architectures offer a single JVM per host machine and keep clusters in sync at the cost of duplicate hardware and increased system administration. As the load increases, these systems must be taken offline to add more hosts, gaining marginal scalability at the cost of availability.

In assessing the scalability of an application server, look for these architectural features:

- A *multi-VM architecture*, with intelligent, multilevel load balancing that matches processing needs for specific operations to Java VMs configured to meet those needs, optimizes application performance and scales further on a single platform. Multi-VM application servers can cost two to 10 times less than single-VM servers in a deployed, high-volume Internet commerce site.
- A *transactional, distributed object cache* that can maintain data integrity efficiently for shared objects across VMs, multiple servers and multiple heterogeneous databases is critical for distributed B2B environments.
- A *persistent object store* can be a performance plus in an application server by providing transparent persistence for Java objects to minimize object-to-relational translation for in-process data and support for distributed, heterogeneous transaction control.

Unify

www.ewavecommerce.com

Availability Beyond Failover

We often confuse high availability with system failover. In some application servers system-level failover is the main availability mechanism; hence, the definition fits. However, mature products offer more efficient failover mechanisms and higher overall availability.

Many systems failover only at the hardware server level. Look for an application server that fails over at the level necessary to maintain service (VM, server, host), reducing the need for expensive hardware redundancy. Look also for tools and features that allow administrators to expand and adapt the e-commerce system without interrupting service. You should be able to add new functionality to your site without taking the Web site, application, application server or hardware server offline. Upgrades or completely new applications should be deployable remotely without downtime.

A highly requested feature of some application servers is the ability to deploy a new version of an application with the same name on the same host while clients are still executing the prior version. New clients are automatically given the new version to use the next time they log in, so upgrades are painless for everyone involved.

Pluggable, Dynamic Security

A complex B2B site requires full-power security at all levels: user accounts, applications and methods, and wire security. Look for application servers with pluggable interfaces for authentication and encryption protocols. These will let you use the latest Java-based security technologies without your having to rewrite or update your applications. For method-level protection look for application servers that offer ACLs (Access Control Lists) for fine-grained and dynamic security.

Pluggable PKI Authentication

Authentication is the process of properly identifying people and things. When a user requests access to your system, the application server must validate his or her identity. *Multifactor* authentication is ideal (see Figure 3). In this approach authentication is based on some combination of who you are (name, user account name), what you have (driver's license, digital certificate) and what you know (mother's maiden name, account password).

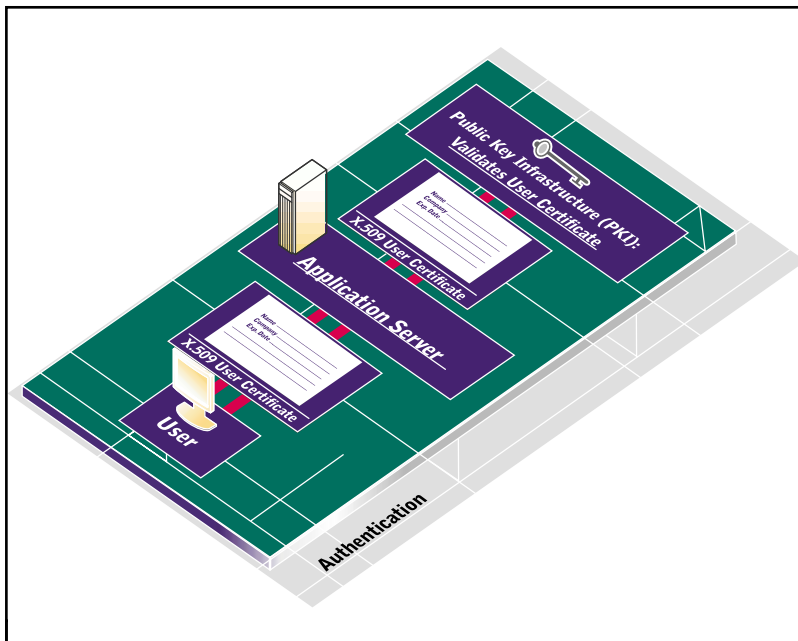


FIGURE 3 Pluggable PKI Security enables multifactor authentication.

User accounts and passwords are already available so a pluggable interface for digital certificates completes the multifactor authentication strategy. PKI (Public Key Infrastructure) is a very strong and versatile authentication mechanism by which certificates are managed and passed to systems and their users. A certificate represents that user throughout the system. The principal (X.509 name) that's attached to the certificate can be associated with a particular VM execution thread, providing a single sign-on by authenticating the user throughout the entire user session.

ACL-BASED AUTHORIZATION

Authorization is the process of determining what actions a user can take and the resources he or she can access. Your application server should allow or deny access based on authorization policies defined by your business requirements.

ACLs are a flexible way of implementing authorization policies. They're objects that can be associated with any application in the system down to the method level, and they allow you to dynamically modify the permissions for any particular user or group. They're explicitly checked by the application being run by the user. When a user or principal attempts to access a system resource, such as an object method or a component, access is authorized or denied based on ACLs associated with that resource. ACLs can also be associated with a named context in a naming service.

PLUGGABLE SSL WIRE PROTECTION

Secure over-the-wire communication is essential for sensitive data transmission such as a funds transfer. As with PKI technology, a pluggable interface will allow you to select the SSL provider that best fits your security needs without changing application code.

AVOID THE PAIN, GO FOR THE GAIN

Okay, so the vendor you're considering has assured you that their application server has all the features you need: a flexible deployment model, scalability to last a lifetime and security enough to satisfy the most demanding paranoid user. Your final question should be, "What do I have to do to get all this?" Do you have to somehow integrate the content services with the business rules engine? Do you need to spend months of developer time adding a nonobject-oriented TP monitor to get distributed transaction control or figuring out how to implement the workflow you need? As far as possible, find an application server that incorporates all the features you need out of the box, either by itself or through true technical relationships with its partners.

Finally, remember that experience counts. Scalability starts with technology but is realized through design. So find a vendor with a proven track record of designing scalable, distributed business systems. They'll help you build an e-business platform that will take you from your first Web application through enterprise and inter-enterprise integration. ☪

osterhaug@worldnet.att.net

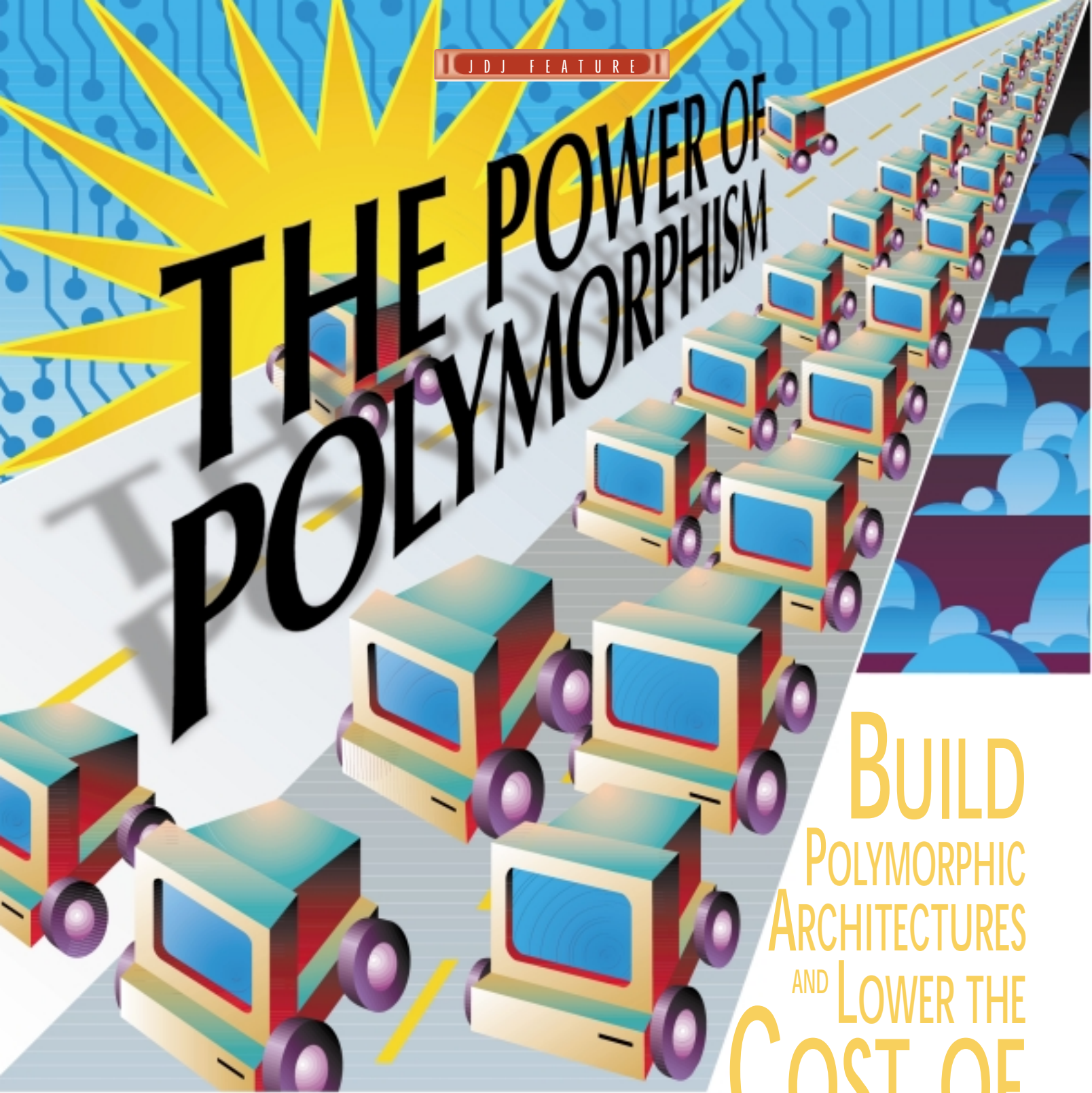
AUTHOR BIO

Anita Osterhaug, a member of the Advanced Application Architecture Team (A3T) at GemStone Systems, Inc., has 19 years of experience in high tech. She's also a professional technical and business writer.

Appeal Virtual Machine

www.jrockit.com

THE POWER OF POLYMORPHISM



BUILD POLYMORPHIC ARCHITECTURES AND LOWER THE COST OF DEVELOPING GOOD SOFTWARE

WRITTEN BY JIM BARNABEE

Polymorphism is an often-misunderstood concept within the developer community. Outside the community it's often a buzzword used to create an image of intellectual capital accumulating in a vault somewhere. This article will attempt to define, in a very basic way:

- What polymorphism is
- What types of polymorphism exist
- How they can benefit the Java developer and architect in building better systems

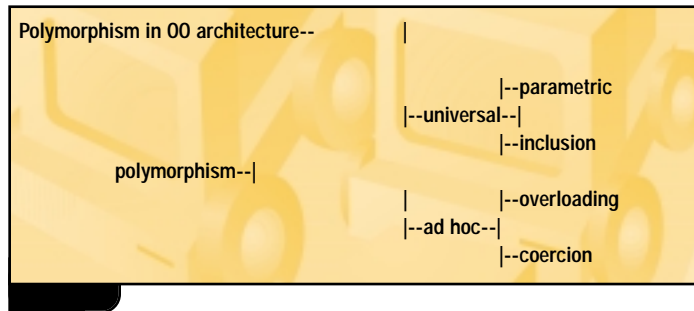
4th Pass

www.4thpass.com

What Is Polymorphism?

Polymorphism is the ability of one object to be treated, or used, like another.

In their ACM report Cardelli and Wegner detailed the types of polymorphism and their relationship to each other. To illustrate this relationship, they created the diagram shown below.



Polymorphism is a powerful tool in the developer's arsenal, allowing architectures to be designed and built that will be flexible enough to change with businesses' needs, yet stable enough not to require redesign and rebuild on a regular basis. Since approximately 80% of the cost of software is for maintenance, building polymorphic architectures can greatly reduce the overall cost of developing good software.

WHY SHOULD I CARE?

Polymorphism is extremely useful in designing enterprise-level systems, as you can create and define multiple interface layers that allow outside parties or outside software to interact with your system in a particular way. We all know how helpful it is in system integration to have a published API, but if interior system changes become necessary, the API may have to be republished and integrated systems may have to be reworked. To avoid this, careful consideration should be given to object APIs as well as the overall system APIs during design. One way to ensure that a robust system can deal with objects that have more complex methodologies than were originally intended is to have the object API be polymorphic. This way, the system knows a "face" of the object being passed to it, but doesn't have to be aware of extemporaneous changes.

Types of Polymorphism

What follows are some simple examples of polymorphism as it relates to a video game environment. These examples are used to further define and illustrate the different types of polymorphism.

OVERLOADING

Overloading polymorphism occurs when a child class overrides the method implementation of the parent class. This type of polymorphism is used when different child classes have different behaviors based on some intrinsic characteristic of the child class. As an example, if milk and vodka are both drink objects, both will have a method called *ingest*. The results of ingesting milk and vodka are extremely different, but when designing a character for a game, the only thing you'd need to be concerned about is ingesting the drink and displaying the results. This is a very simple example of overloading polymorphism, since each subclass is redefining the *ingest* method to return results for its particular class type (see Listing 1).

INCLUSION

Inclusion polymorphism occurs if a child class inherits its method substance from the base or parent class. This enables objects or systems that would previously have used the base class to use the child classes with equivalent results. Inclusion polymorphism is most useful in masking the inherent complexities of child classes from objects or systems that previously dealt with the parent class. As an example, if the drinks in question

were water and Perrier, they might share the *ingest* method from the base class of drink. This would be an example of inclusion or subclassing polymorphism. Regardless of whether the polymorphism in this example is overloading or inclusion, your character in the game only needs to be concerned with ingesting the drink and dealing with the results.

```
public Class Water extends Drink
{
    public Water(){}
}
public Class Perrier extends Drink
{
    public Perrier(){}
}
```

Overloading and inclusion are the most common types of polymorphism in Java (along with special cases of overloading polymorphism that relate to interfaces, which we'll discuss later). Parametric and coercion polymorphisms are also used in Java, but to a lesser degree in most implementations.

Relationship Between Inclusion and Overloading Polymorphism

Inclusion polymorphism exists when several subclasses use the method of the superclass (parent) to perform whatever action is required.

Overloading polymorphism exists when each subclass defines its own method of action, either because the parent class has declared the method abstract, or because it simply wishes to provide special processing.

PARAMETRIC

Parametric polymorphism occurs when a class or classes implement methods that are the same in signature except for the parameters passed to the method. This is extremely useful, as one class can handle a great many different types of arguments to a specific method, enabling the class to be used as a bridge between the different types of objects that wish to communicate with your system and the system itself. An example of parametric polymorphism would be if an *ElectromagneticRailGun* class had multiple methods for *fireProjectile* – say, one that took a chunk of iron ore and one that took a missile. The results of firing the projectile would be very different depending on the type of projectile that was fired.

```
public Class ElectromagneticRailGun
{
    public String fireProjectile(IronOre chunk)
    {
        return new String("Boy that smarts!");
    }
    public String fireProjectile(Missile icbm)
    {
        return new String("Global Thermonuclear War");
    }
}
```

COERCION

Coercion polymorphism occurs when a primitive or object type is cast or "coerced" into being another primitive type or object type. An example of coercion polymorphism would be if the character display in the game always showed the number of rounds left in the current weapon, but an energy gun might actually measure the amount of charge left in less than one unit increments, say, 42.3. A revolver, on the other hand, might show five bullets left. But the display only shows whole numbers. So, when updating the display for the energy weapon, we might use casting to change the float value of 42.3 to an integer value of 42. This is usually inefficient, and always dangerous since you're losing precision, but it's sometimes necessary.

Fiorano

www.fiorano.com

```
public Class Display extends java.awt.Label
{
    public Display(){super();}
    public setBulletsLeft(int in)
    {
        super.setText(String.valueOf(in));
    }
    public setEnergy(float units)
    {
        setBulletsLeft((int)units);
    }
}
```

This is usually a really bad idea, and may result in casting exceptions thrown at runtime. A safer way to convert primitive values in Java is to use the conversion interfaces defined with the language itself. The code above can be rewritten to include this methodology as follows:

```
public Class Display extends java.awt.Label
{
    public Display(){super();}
    public setBulletsLeft(int in)
    {
        super.setText(String.valueOf(in));
    }
    public setEnergy(float units)
    {
        public Float f = new Float(units);
        setBulletsLeft(f.intValue());
    }
}
```

Object coercion is a larger and more difficult subject, and won't be detailed here. If a Class coercion situation occurs, consider using the instanceof keyword to provide safer, more stable implementations.

```
public class Processor
{
    public Processor(){}
    public void process(Parent p)
    {
        if (p instanceof Child1)
        {
            //do something
        }
        else
        {
            //do something else
        }
    }
}
```

Multiple Inheritance vs Interfaces

In Java there's a special type of overloading polymorphism to examine. It comes about because of the addition of interfaces to the Java language. An interface is usually defined in a separate file, and specifies the API of the object that will implement the interface.

When Java was first introduced, a lot of C++ programmers were concerned by the lack of multiple inheritance in Java. In some OO languages, such as C++, multiple inheritance is allowed. In Java we have only single inheritance, that is, any class can extend only one parent class. A lot of people starting out in Java see this as a handicap, but they usually change their minds once they understand the power of interfaces.

It's possible to use implementation of interfaces to achieve multiple inheritance. Interfaces in Java don't really allow multiple inheritance, since all classes that implement an interface must implement all methods declared by the interface, just as all classes that subclass a parent must implement all abstract methods declared in the parent class or declare them abstract themselves. Interface implementation does allow for a rather unique type of inclusion polymorphism, however. Since any method can take an interface type as an argument, the method can always be sure that whatever kind of object it's receiving will have all the methods declared in the interface, and they will always return the required type of response. Since a class can implement multiple interfaces, the ability of any object to be utilized by multiple methods increases geometrically as the number of interfaces it implements and the number of methods that accept that interface as a parameter type increase.

To use our video game example, molten metal could implement the drinkable interface just as the drink class does, but the results would be drastically different.

```
public Class MoltenMetal implements Drinkable
{
    public MoltenMetal();
    public String ingest(){return new String("ouch!!");}
}
```

Parametric Polymorphism in E-Commerce Situations

Often in building a fairly robust system you want to expose parts of the API to the general public. You might wish to have logging methods that took the object where the error occurred, or a String containing the error message to be stored or an Exception that would denote what kind of problem occurred. These types of method signatures might look something like:

```
static public void log(Object o, int level)
static public void log(String o, int level)
static public void log(Exception e, int level)
```

So sending data to the log, with either the object causing the problem (or a message about the problem) or the Exception that was generated would yield applicable results.

INCLUSION POLYMORPHISM IN E-COMMERCE SITUATIONS

Inclusion polymorphism is used in e-commerce situations to design and build robust systems that don't require major modification when the highly volatile market changes. In designing and building a transaction server, you might want to build one that processes Transaction Data Objects (TDO). But you might need to create new types of TDOs depending on the type of transaction you wished to process. The beauty of inclusion polymorphism in this situation is that the transaction server code would never change, and any class that extended TDO could be passed to the transaction server for processing.

```
public Class TDO
{
    public TDO(){
        //some transaction methods
    }

    public class TransactionServer
    {
        public TransactionServer(){
            public void process(TDO trans)
            {
                //transaction processing code
            }
        }
    }
}
```

Software AG

www.softwareag.com/bolero

00P.

WWW.00

.com

op.com

North woods

www.nwoods.com

Object- wave

www.objectwave.com

OVERLOADING POLYMORPHISM IN E-COMMERCE SITUATIONS

As discussed earlier, interfaces allow for a special type of overloading polymorphism in Java. This becomes extremely useful when we consider that we've built our transaction server and now we want to run it in a CORBA environment. We can use the declared interfaces of our services and extend the base classes supplied by (or created via a tool by) the ORB vendor, and we've got instant CORBA-compliant objects.

Overloading is also useful if you have a specific type of processing that each subclass needs to do, but a lot of "overhead" processing that all classes of that type need to do. As an example, you can declare an abstract method called *process* in your Base servlet class, which gets called whenever you get a Post request. When you extend the Base class, all the child classes can then process the request in their own way. The base class can contain all the methods to set the return type, put headers and footers into the response, set up connections, read parameter files and do all the other "housekeeping" chores. That way, when you need to create a new servlet, you simply extend the base class and implement one method containing your business logic.

COERCION—THERE'S USUALLY A BETTER WAY

Coercion has limited use in most production environments, due to the dangers of losing precision and possibly a ClassCastException at runtime. If you often find yourself using casting, try using the object translation methodologies mentioned previously or try creating the most generic situation possible, then using the *instanceof* keyword to determine the actual type of object you're dealing with for specific processing.

This article is intended as a general overview of polymorphism and should by no means be considered definitive on the subject. For further information and mathematical constructs detailing polymorphic structures in object-oriented design methodologies, refer to L. Cardelli and P. Wegner's "On Understanding Types, Data Abstraction, and Polymorphism" (ACM Computing Surveys, vol. 17(4).) ☉

AUTHOR BIO

Jim Barnabee is a Java architect whose main focus is designing and developing Java clients and servers for Web-based solutions to business problems. Jim has five years of Java experience (beginning with the 0.1 beta) and has been programming professionally since his junior year of high school. Jim routinely publishes online and print articles on Java development topics.

mutabletao@yahoo.com

Listing 1

```
public Class Drink implements Drinkable
{
    public Drink(){}
    public String ingest(){return new String("drank");}
}

public Class Milk extends Drink
{
    public Milk(){}
    public String ingest()
    {
        return new String("Does a body good");
    }
}

public Class Vodka extends Drink
{
    public Vodka(){}
    public String ingest()
    {
        return new String("You are now tipsy");
    }
}
```



Pramati

www.pramati.com

THE USE OF JNDI IN ENTERPRISE

A common framework for locating enterprise computing components

JAVA APIs

WRITTEN BY IAN MORAES

Enterprise system developers have embraced the Java 2 Platform, Enterprise Edition (J2EE) development model and the collection of Java APIs for developing standard, component-based, multi-tiered enterprise solutions. These APIs provide a standard mechanism for accessing pertinent system services typically required for enterprise systems development such as databases (JDBC), asynchronous communication (JMS), transaction support (JTA), e-mail (JavaMail/JAF), distributed computing (RMI/IIOP), naming and directory services (JNDI) and Web client presentation (servlets/JSP). Enterprise JavaBeans (EJB) is a core J2EE API that provides for a standard server-side component model.



JNDI is an enterprise Java API that provides naming and directory functionality using a common set of classes and interfaces that are independent of a specific naming and directory service implementation. A standard extension to the Java platform, some developers consider its use to be limited to interfacing with directory services such as LDAP. However, this narrow view fails to fully appreciate its flexibility and power. JNDI is a particularly important J2EE API given its pervasive use among other important enterprise Java APIs such as EJB, JDBC and JMS. This article introduces JNDI as a common framework for locating enterprise computing components using enterprise Java APIs such as EJB and the Java Message Service (JMS).

IAM

www.iamx.com

Naming and Directory Services

Before discussing JNDI, a quick overview of some naming and directory service terminology might be helpful. Although there are a number of existing naming and directory services, they all share some common concepts and terms that are helpful in understanding the flexibility of JNDI. Here's a list of some well-known naming and directory services:

- **LDAP (Lightweight Directory Access Protocol):** A directory services protocol that enables clients to manage and query a hierarchical repository of entries and attributes. For example, LDAP can be used to locate information pertaining to users (e.g., X.509 certificates) and network resources (e.g., available printers).
- **NIS (Network Information System):** A naming service, formerly known as yp, developed by Sun. It allows users to access systems on other hosts with a single user ID and password.
- **COS Naming Service:** The CORBA naming service that enables CORBA clients to locate CORBA server objects using names.
- **DNS (Domain Name Service):** A distributed naming service that allows users to refer to hosts using easily remembered names (e.g., anthony) instead of IP addresses.

A naming system can be described as a mechanism that associates names with objects and provides a way to find an object based on a given name. A binding can be described as an association of a name with an object. A file name, for example, is associated with a reference that an application can use to access a file. A binding is defined relative to a context. A context comprises a set of bindings. For example, /home, the UNIX file directory, is a context comprising a set of bindings pertaining to files and directories. The process of looking up an object in a context based on a name is known as *resolution*. This would be loosely analogous to finding a phone number of a person by looking up his or her name.

All objects in a context are named using a specific naming convention. Thus a naming system can be described as a set of contexts with the same naming convention. For example, in file systems a name in the DOS file system (c:\boot.ini) has a different naming syntax than a name in the UNIX file system (/etc/gated.conf). In contrast, LDAP's naming syntax reads names from right to left. An example of an LDAP name is "cn=Ian, o=glenayre, c=us".

When discussing naming systems, we need to describe different types of names. An *atomic* name is simply an indivisible component of a name. For example, in the directory name /etc/mail, mail is an atomic name. A *compound* name is a sequence of atomic names that conforms to a specific naming convention. For example, /var/adm is a compound name. *Composite* names span multiple naming systems. For example, a URL such as http://www.glenayre.com/index.html spans different naming systems such as DNS (Domain Name Service), and the UNIX file system. *Namespace* is a term used to describe the collection of names in a naming system.

A directory service is an extension of the naming service. A directory is typically used to associate attributes with objects. For example, a user object could have attributes such as the user's name, phone number, e-mail address and fax number. Directory services provide functionality for evaluating and modifying attributes attached to contexts and the ability to search a context using those attributes as a filter.

Although naming and directory services such as LDAP, NIS and DNS vary in their exact implementation from one service to the next, JNDI provides a common framework for accessing existing heterogeneous naming and directory services in a way that hides their protocol and implementation details.

JNDI Architecture

The JNDI architecture can be described in terms of three main layers. JNDI's layered architecture enables clients to use a common API to work with different naming and directory services (DNS, LDAP, COS Naming).

The top layer is the application layer that uses the JNDI API. The middle layer is the JNDI API that defines a set of classes and interfaces for sup-

porting common naming and directory service functionality. This is the layer that frees a developer from having to deal with specific naming and directory service complexities. It relies on concrete implementations of naming and directory services.

The bottom layer is the implementation layer. Since the JNDI API is independent of specific naming and directory services, it's up to service providers to provide implementations using the JNDI Service Provider Interface (SPI), which can access different naming and directory services. These service provider implementations can easily be plugged in to the JNDI architecture, thus enabling the application layer to access their services using the JNDI API. Figure 1 shows an overview of the JNDI architecture.

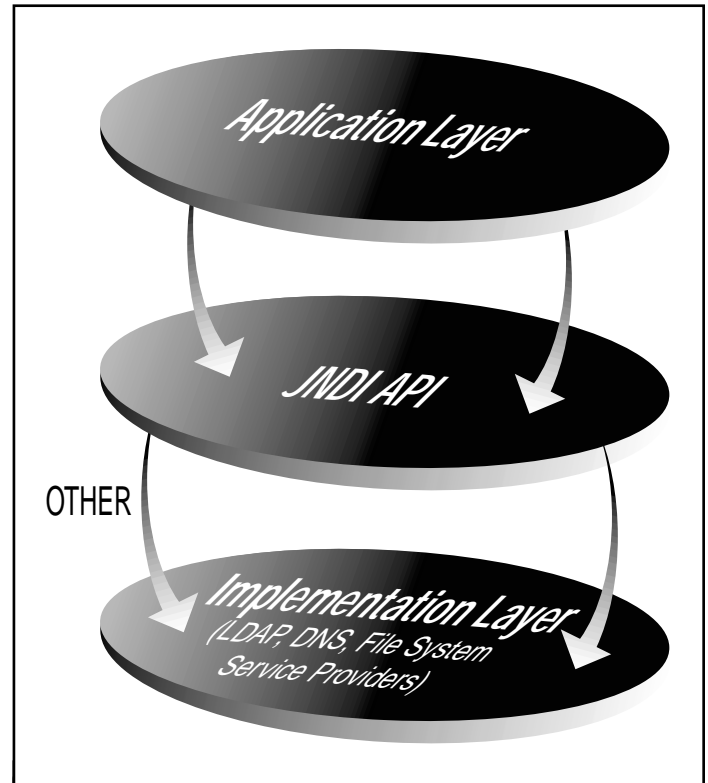


FIGURE 1 Overview of JNDI architecture

The JNDI API comprises a number of packages. To provide a more concrete description of the JNDI architecture, the main JNDI packages are summarized below.

- **javax.naming package:** Defines classes (e.g., Binding) and interfaces (e.g., Name) for accessing naming services
- **javax.naming.directory package:** Defines classes (e.g., InitialDirContext) and interfaces (e.g., Attribute) for accessing directory services
- **javax.naming.event package:** Defines classes (e.g., NamingEvent) and interfaces (e.g., NamingListener) for event notification when accessing naming and directory services
- **javax.naming.ldap package:** Defines classes (e.g., InitialLdapContext) and interfaces (e.g., ExtendedRequest) to provide support for LDAPv3 extended operations and controls
- **javax.naming.spi package:** Defines classes (e.g., NamingManager) and interfaces (e.g., InitialContextFactory) used to access existing naming and directory services using JNDI's SPI

Now that you have a basic overview of JNDI's architecture, we can explore some main JNDI classes and interfaces needed to support a JNDI client. Table 1 describes some fundamental JNDI classes and interfaces.

Note that if you're using the JNDI API within a multithreaded application context, you may need to develop your own locking mechanisms because instances of some JNDI classes you may use (e.g., Binding) don't support concurrent access.

SIC

www.sic21.com

CLASS/INTERFACE NAME	DESCRIPTION
Name	This <code>javax.naming</code> interface represents a name. The <code>CompositeName</code> class represents a sequence of names from multiple namespaces. The <code>CompoundName</code> class represents a sequence of name components from a namespace where each component is an atomic name. Both classes implement the <code>Name</code> interface.
Binding	This <code>javax.naming</code> class represents a name-to-object binding in a context.
Context	This is a core <code>javax.naming</code> interface that represents a naming context that comprises a set of bindings. It has methods for adding a binding, listing bindings, modifying bindings and looking up an object using a name.
InitialContext	This <code>javax.naming</code> class implements the <code>Context</code> interface and provides an entry point for clients to perform naming and directory operations.
DirContext	This <code>javax.naming.directory</code> interface extends <code>Context</code> and is a core directory interface that has methods for searching a directory, and exploring and updating attributes associated with a directory object. A directory object can comprise a number of attributes.
Attribute	This <code>javax.naming.directory</code> interface represents a single directory attribute associated with a named object. The <code>Attributes</code> interface represents a set of attributes associated with a directory entry.
NamingEnumeration	This <code>javax.naming</code> interface represents enumerating lists returned by operations in the <code>javax.naming</code> and <code>javax.naming.directory</code> packages.
NamingException	This <code>javax.naming</code> class is the superclass of all exceptions thrown by naming and directory operations.
Reference	This <code>javax.naming</code> class represents an external object reference (i.e., in a different naming/directory system) and facilitates cooperation among different naming systems.

TABLE 1 Some JNDI classes and interfaces

Explore Your File System Using JNDI

To introduce you to some of the JNDI classes and interfaces, I'll use a simple and familiar use case of getting a listing of a directory in a local file system.

The first task is to obtain a reference to an `InitialContext` so as to provide an entry point to a naming system. Before accessing naming and directory services, JNDI client applications can specify environment properties (e.g., security preferences, context factory class name) of a context. A context's environment is represented as a hashtable that's used in the constructor of `InitialContext`. Note the use of a file system provider's factory class name for creating the initial context and the use of a URL to specify the file system URL (e.g., `file:///`). This type of environment specification detail could be encapsulated in a factory method or facade depending on your application's design.

```

Hashtable contextEnv = new Hashtable();
contextEnv.put(Context.INITIAL_CONTEXT_FACTORY,
"com.sun.jndi.fscontext.RefFSContextFactory");
contextEnv.put(Context.PROVIDER_URL, directoryURL);
Context context = new InitialContext(contextEnv);

```

After obtaining a reference to a context, we can get a list of bindings from the context. From the binding we can obtain a name of an entry in the file system's directory.

```

try {
    NamingEnumeration bindingList = context.listBindings("");
    while (bindingList.hasMore()) {
        Binding binding = (Binding) bindingList.next();
        System.out.println(binding.getName());
    }
} catch (NamingException ex) {
    ex.printStackTrace();
}

```

Although security properties (e.g., identity of principal, security level) can be specified in the environment properties prior to connecting to a naming or directory service, JNDI doesn't specify a specific security model for accessing naming and directory servers. Security-related operations are the responsibility of individual service providers. However, JNDI can be used to query a resource (e.g., LDAP server) to determine the security mechanisms (e.g., Simple Authentication and Security Layer [SASL] mechanisms) supported. In addition, JNDI does define security-related exceptions such as `javax.naming.AuthenticationException`, which are thrown when a user can't be authenticated by a naming or directory service.

Use of JNDI in EJB

Now that you've seen how some of the main JNDI classes and interfaces are used, let's discuss the role of JNDI in the EJB specification. EJB is a standard component architecture for developing distributed applications using Java. A naming service is an important distributed computing system service that enables a client to obtain a reference to a distributed object using a name. For those who have developed systems using CORBA, you've probably used the CORBA naming service or an ORB vendor-specific mechanism to obtain a reference to a CORBA object.

For EJB systems JNDI is used to obtain references to an EJB's home interface, a factory interface for creating, finding and removing EJBs. In addition, JNDI can be used to gain access to an EJB's environmental entries, which are used to customize an EJB's behavior. I'll discuss these two uses of JNDI in the EJB 1.1 specification.

LOCATING EJB HOME INTERFACES

An EJB client can locate an EJB's home interface through the standard JNDI API. The EJB container is responsible for making the home interfaces of its deployed beans available to an EJB client using JNDI. A client's JNDI namespace may be configured to include the home interfaces of EJBs that are installed in containers on other hosts.

Consider the use case of an application that provides messaging and informational services (e.g., messages, stock quotes, news). To support

MetaMata

www.metamata.com

this client, assume we have a coarse-grained distributed façade in the form of a ContentMessageBroker session bean. To obtain messages and content (e.g., stock quotes), we need to obtain a reference to the ContentMessageBroker's home interface. Following is a code snippet to show how a messaging and informational services client obtains a reference to the ContentMessageBroker's home object.

```
try {
    InitialContext = new InitialContext();
    Object obj = initialContext.lookup("ContentMessageBrokerBean");
    brokerHome = (ContentMessageBrokerHome) PortableRemoteObject.narrow(obj,
        arch40.proto.ContentMessageBrokerHome.class);
} catch (Exception nameException) {
    nameException.printStackTrace();
}
```

A useful development practice is to wrap the code shown above in a Façade class. This client-side façade can shield your application's business layer from specific distributed computing configuration and remote object access details (e.g., to obtain a reference to a session bean). For more information on the Façade design pattern consult Gamma et al.'s *Design Patterns* (1995).

As we discuss the use of JNDI in distributed systems, it's appropriate to discuss load balancing, a requirement of most large-scale enterprise systems. A number of load-balancing strategies are available, including IP-level load balancing and DNS "round robin"; for CORBA systems the naming service can be used to provide a load-balancing solution. For EJB systems JNDI has been used to provide a proprietary load-balancing solution for an EJB Server in the form of a JNDI service provider that uses an underlying naming service. Thus this service provider implementation is used to select an object from a cluster of servers using JNDI interfaces.

ACCESSING EJB ENVIRONMENT CONTEXT

An attractive feature of component-based development is the ability to customize a component's behavior without having to change its source code. An EJB can be customized using environmental entries specified in the EJB deployment descriptor. The EJB container provides an implementation of a JNDI context that comprises the EJB's environment and makes it available to an instance of that EJB at runtime using the JNDI. This use of JNDI is another way for an EJB to interface with its container. Further, an EJB container must be able to save and restore a reference to this JNDI context across EJB instance passivation.

Suppose we consider a contrived example of a session bean that maintains a configurable number of connections to legacy system servers. Let's assume we need to be able to configure the maximum number of connections during deployment. First, the EJB finds the environment naming context using JNDI by creating an InitialContext object. The EJB then looks up the environment naming context using the name java:comp/env. The maxConnections environment entry is declared in the env-entry element in the bean's deployment descriptor – in the EJB 1.1 specification, deployment descriptors are specified using XML. The following code snippet shows how a session bean instance can obtain the maximum number of connections defined during deployment.

```
// Obtain the EJB's environment naming context
Context initContext = new InitialContext();
Context legacyServerContext =
(Context)initContext.lookup("java:comp/env");
// Obtain the maximum number of server connections configured by the
Deployer.
Integer maxConnects = (Integer) legacyServerContext.lookup("maxConnections");
```

Use of JNDI in JMS

JMS is a J2EE API that provides support for interfacing with enterprise messaging systems using two messaging models: point-to-point and publish/subscribe. The intrinsic value of JMS, JavaMail, JNDI and other enterprise Java APIs is that they provide a common interface to existing and disparate enterprise system APIs (mail systems, naming and directory systems, and so on).

JMS defines a specific type of object called *administered* objects that constitutes JMS configuration information. These objects, such as ConnectionFactory, are created by a JMS administrator and registered so they can be globally accessed by JMS clients using JNDI. ConnectionFactory encapsulates a set of configuration parameters that have been defined by an administrator and is used by a JMS client to create a connection with a particular provider.

JNDI is the convention used by JMS clients to look up JMS administered objects. If an administrator has created and configured a TopicConnectionFactory, for instance, it can be found as follows using JNDI:

```
Context umsMessages = new InitialContext();
TopicConnectionFactory connectionFactory =
    (TopicConnectionFactory) ctx.lookup("javax.jms.TopicConnectionFactory");
```

The ConnectionFactory can then be used to create a Connection object that is used to interact with an existing messaging system. The use of JNDI in JMS is similar to its use in JDBC – JNDI is used to retrieve references to DataSource objects in order to obtain a database connection. For an introduction to the Java Message Service, visit the JMS Web site listed under "Resources" at the end of this article.

Conclusion

The JNDI API is a powerful API used by other enterprise Java APIs. For those interested in exploring JNDI, version 1.2 is currently available from the JNDI Web site listed below. The JNDI 1.2 class libraries and the COS naming service provider are packaged as part of the Java 2 SDK, Enterprise Edition.

This article did not focus on a core value of JNDI: its ability to access heterogeneous directory services using a common interface. The JNDI Web site lists available service providers for directory services. On a related note, if you're developing systems that use disparate directory services, you might also be interested in exploring Directory Services Markup Language (DSML), an XML language. DSML is an emerging effort to represent directory information using a format that's independent of a specific directory service. A preview release of a JNDI service provider is available from the JNDI Web site. ☺

Resources

1. *Java Naming and Directory Interface*: www.javasoft.com/products/jndi
2. *Enterprise JavaBeans*: www.javasoft.com/products/ejb
3. *Java Message Service*: www.javasoft.com/products/jms
4. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
5. *Directory Services Markup Language*: www.dsml.org

AUTHOR BIO

Ian Moraes, Ph.D., is a principal engineer at Glenayre Electronics where he works on the architecture and design of a unified messaging system. Ian has developed client/server systems for both the financial services and telecommunications industries.

imoraes@atlanta.glenayre.com

iona

www.iona.com

J2EE and WML: Holding a Royal Flush

Will you hold the winning hand?



WRITTEN BY
JASON WESTRA

The buzz at JavaOne 2000, in my opinion, was definitely the solidification of Java in the wireless market. As radio host for SYS-CON Radio at JavaOne, I had the pleasure of interviewing CEOs and CTOs of leading application server vendors. Many of them focused, not on J2EE support, but on how their products are providing wireless solutions.

Wireless access to EJBs is becoming a reality due to support for WML in numerous application servers. BEA's WebLogic Server and Lutris's Enhydra are two of the players already shifting attention to wireless enhancements to their products, building on the core J2EE services that offer reliability, scalability and transaction support to wireless business solutions.

This month in **EJB Home** I'll cover some basic concepts of wireless development and describe why I think using a WAP-enabled application server over the J2EE platform is a winning hand.

Wireless Application Development

What is WAP? Good question. Let's discuss wireless concepts and common terminology, then apply it to J2EE application servers to understand the importance of EJB and the rest of the J2EE APIs in wireless development.

WAP - Wireless Application Protocol - is a set of protocols designed to allow the development of Web-based applications for mobile devices such as phones and handheld computers. WAP was developed by a group of industry giants including Motorola, Ericsson and Nokia in a fashion similar to the way Sun requests representatives from industry leaders to help develop new Java specifications. The WAP Forum, as they're called, developed WAP to address numerous limitations in the mobile applications. For instance, mobile devices have limited CPU and memory compared to a standard desktop computer. WAP is designed to overcome these limitations. Likewise, the reliability of communications for mobile

devices is unpredictable at best, while limited bandwidth generally causes network latency. These constraints to successful wireless solutions are also the focus of the WAP Forum.

Two elements that resulted from the WAP Forum are an application protocol for wireless communication and a browser-based application environment called WAE (Wireless Application Environment). Together they provide the capability to program wireless solutions for most mobile devices.

Wireless Application Environment

WAE provides two components that enable the development of wireless applications that are network-neutral and communicate with narrow-band devices: Wireless Markup Language (WML) and WMLScript (WMLS).

WML is similar to HTML, which is passed between a browser and a Web server via the HTTP protocol. It's an XML-based language, specifically designed to provide a presentation layer in mobile devices with microbrowsers. If you think HTML clients are basic, wait till you work with (or develop) a microbrowser UI!

The WAP Forum developed a specification that defined the tags and format of a well-formed WML document. A WML document comprises one-to-many cards that in turn are contained in a deck when they're logically grouped together to perform a business function. Cards are the lowest form of UI presentation in a microbrowser. They generally map just enough information to display on one screen in a mobile device. That said, more steps are generally involved in a wireless business transaction than in traditional Web

browser applications. More steps alone tend to signify less reliability for wireless solutions. The need for solid business logic and transaction control in your application server becomes even more evident with wireless application development.

WMLScript, the other enabling element of WAE, provides general scripting capabilities to overcome the narrow-band communication limitations between mobile clients. Together, these elements are vital pieces in a wireless solution to minimize the load on narrow-band devices. For instance, a WAP Forum study comparing the same application based on an HTTP protocol and the WAP protocol found less than half the number of packets are required for WAP as for HTTP.

Let's look at the mechanism for encoding and decoding WML between mobile devices and an application server.

WAP Gateways

WAP Gateways are products that provide an implementation of the end-to-end communications for WAP. A WAP Gateway connects clients on a mobile network with J2EE application servers that have been configured to talk with the WAP Gateway. WAP Gateway interoperability is the latest and greatest feature that J2EE application servers are offering to make business services built on J2EE APIs accessible to mobile devices. WAP Gateways act as bridges between mobile clients requesting business services and the application servers housing the logic and data. Figure 1 shows this relationship in more detail.

KL Group

www.klgroup.com/power

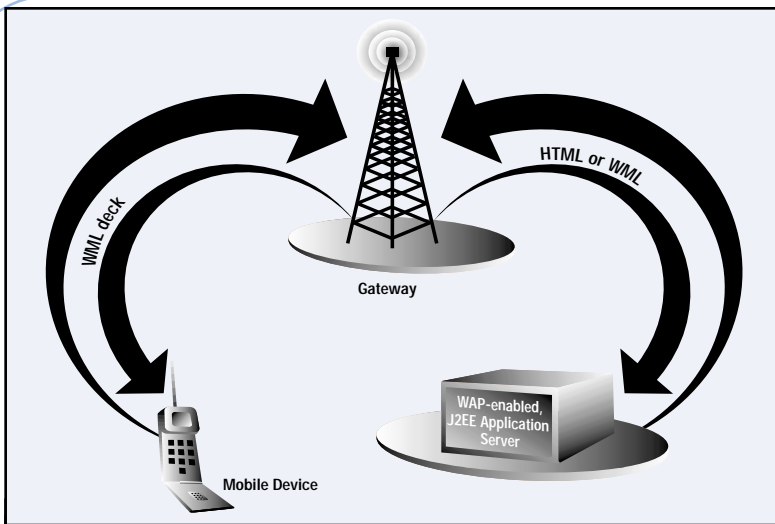


FIGURE 1 WAP Gateway

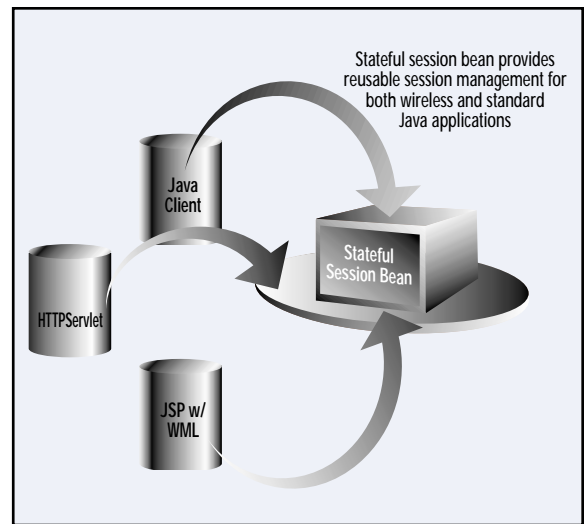


FIGURE 2 Reusable session management

J2EE Vendors Gone Wireless?

Wireless support has just begun in J2EE application servers. Perhaps it seems late in the game, considering the WAP Forum has been around since 1997. In any event, I see vendors providing support for standards based on market demand and technological feasibility. Basically, the Java market demanded J2EE support; thus vendors put the inclusion of J2EE APIs at the top of their list. EJB, the focal point of the J2EE platform, has been available since 1998, and Java industry giants from IBM, BEA Systems and Oracle on down have enthusiastically adopted its APIs (although some still hold out on J2EE licensing issues with Sun). These same companies are joining the WAP Forum in the golden age of wireless. Their servers are including wireless support on top of solid J2EE underpinnings.

WAP, a relatively new standard, is increasing in popularity not only as a result of market demand, but also because it's maturing into the de facto standard for wireless communications. Application server vendors can now rest easy, knowing their product costs to support WML won't go to waste but will remain standard for years to come. The fear of writing an outdated product line is much higher when you don't build it on industry-supported standards. It's hard to justify the resources and time for an effort until there's enough momentum for the technology to provide an enduring market.

Advances in wireless technology are affecting the proliferation of the wireless applications market. Mobile devices are packing more memory and processing power these days. Also, the bandwidth for communications is increasing enough to support shuttling more and more com-

plex data into mobile devices. WML supports tags for graphics and icons that can be downloaded into microbrowsers, a huge step in bandwidth capabilities.

J2EE: The Underpinnings of Scalable, Reliable Wireless Solutions

As indicated earlier, EJB is a more mature standard than the wireless standards emerging on the market. Since EJB is now inherent in most app servers, and J2EE is a stable platform for building mission-critical applications, it's an easy transition to build wireless capabilities into application servers to take advantage of existing and new business logic built with EJBs.

Core business logic is being built with EJB and the J2EE platform today. Companies are taking advantage of a platform that is open, standards based and powerful with respect to scalability and transaction processing. These key features are exactly what wireless applications need to provide users with an enjoyable experience. Let's take a closer look at two J2EE features: session and transaction management, and scalability, to understand why J2EE application servers play an important part in a winning hand called wireless application solutions.

SESSION AND TRANSACTION MANAGEMENT

"I pick my first two cards dealt by the man in the cheap tuxedo and see the Ten of Hearts and the Jack of the same suit..."

Because wireless environments tend to have less connection stability and less predictable availability of service, development of wireless applications must take these constraints into account. Reliability of session data and transactions is a strong suit of the J2EE platform and an added benefit to wireless applications built on a J2EE app server.

J2EE, centered on EJB, provides the transaction management that wireless applications need, especially when their communications are even less reliable than an HTTP session between browser requests. Applications looking to take advantage of EJB's session and transaction management features might build their business process layer with stateful session beans. Then, when a wireless device drops its connection, a transactional, stateful session bean allows the device to regain its transaction where it left off and continue processing as soon as it connects again.

There are many ways to use the session management features of EJB for wireless applications. The beauty of developing business logic with EJBs is that this logic can be reused for other front ends. In the example I described earlier, you can use a stateful session that manages a complex process that may span across multiple WML decks in order to allow the user of a mobile device to participate in a long-running business transaction. However, the business process modeled by the stateful session bean shouldn't be based on the user interface you're trying to support. Instead, it should reflect your business, and it should be just as easy to paste a JSP or servlet-driven UI onto the front of the session bean's logic and data as it is to develop a WML-based interface (see Figure 2).

SCALABILITY

"My third card is the Queen of Hearts. Card number four...the King! And last but not least, I hold the Ace of Hearts in my hand - an unbeatable royal flush."

The following quote caught me by surprise last year....What's your reaction? "According to The Strategis Group, there will be more than 530 million

AUTHOR BIO

Jason Westra is the CTO of Verge Technologies Group, Inc. (www.vergecorp.com). Verge is a Boulder, Colorado-based firm specializing in e-business solutions with Enterprise JavaBeans.

Parasoft

www.parasoft.com

the XML
DevCon

2000-2001

Conference
Series

THE

FUTURE

HAS ARRIVED

SYS-CON Media and Camelot Communications are pleased to announce our schedule of upcoming XML DevCon conferences. Each event is a unique combination of superior technical sessions and interactive exhibitions designed specifically for XML enthusiasts. If you're serious about accelerating your XML programming skills, plan to join us.

Via print publications,
online resources
and live events, we're
all you need to remain
well informed on
XML and every other
emerging technology.

XML
DevCon
FALL
2000

November 12-15, 2000
Doubletree San Jose Hotel
San Jose, California

XML
DevCon
EUROPE
2001

February 19-21, 2001
Novotel-London West
London, England

XML
DevCon
2001


Spring/Summer 2001
New York City

XML
DevCon
NOVEMBER
2001

November 2001
San Jose, CA

WWW.SYS-CON.COM or INFO@CAMELOT.COM.COM

Presented by:  

Produced by: 


Wireless
DevCon

LEVERAGING
BUSINESS
IN THE

UNWIRED
WORLD

CONFERENCE
December 3-5, 2000

Doubletree San Jose Hotel
San Jose, CA

www.WIRELESSDEVCON2000.com

**Early Bird
Registration:**

Register before
October 27th
and **SAVE \$200!**





E J B H O M E

wireless subscribers by the year 2001. New estimates report that the number of wireless subscribers will break the one billion mark by 2004, and a 'substantial portion of the phones sold that year will have multimedia capabilities.'

If the Strategis Group estimate is correct and J2EE continues to grow as the preferred platform for server-side computing, the number of users connected over the Internet to J2EE-based application servers will be mind-boggling!

No worries. J2EE application servers are built with highly scalable distributed architectures in mind. They are built to manage thousands of client connections, threads and transactions at a time. Servers on the market today are also able to scale relatively well in stateful architectures that maintain session across multiple calls from a client. Stateful architectures will be essential to successful wireless applications as duly noted above under the "Session and Transaction Management" section.

Scaling to support unknown, potentially unlimited users across the Internet is a problem already being addressed in many Web-based applications today. However, the wireless movement opens up the Internet to a whole new market of users. They are the "anytime, anywhere" users whose surfing time isn't governed by a DSL line and whose business doesn't wait for a LAN connection. You must begin to think about building your applications based on the "anytime, anywhere" user, including scaling and transaction management for wireless solutions.

Conclusion

This month I covered a hot topic in the area of Java these days: wireless application development. With a little background on the WAP, the de facto standard for wireless development, I tied the J2EE platform into the picture to magnify the importance this platform plays in wireless solutions. The demand for wireless applications is putting even greater pressure on the J2EE platform to rise up and meet the scalability, session management and transactional needs of the wireless community.

If you pick a J2EE-compliant application server as your deployment environment, you're playing a royal flush when it comes to the success of your wireless application. I believe the wireless solutions market will be dominated not only by innovative device manufacturers, but by application server vendors who make it easy to build reusable, scalable business logic that supports the smoke and mirrors of WML decks and cards. They who host their wireless applications in a scalable, transactional fashion hold the winning hand! ☛

jwestra@vergecorp.com

JAVA DEVELOPERS JOURNAL.COM



2000

READERS' CHOICE
AWARDS



Go
Online
and
Subscribe

Today!
SYS-CON.com

or
Call
1-800-513-7111

Subscribe to the
Finest Technical
Journals in the Industry!

GET
YOUR
OWN!



FUAT KIRCAALI, PUBLISHER, JAVA DEVELOPER'S JOURNAL

FROM THE PUBLISHER

Introducing... The *JDJ* Approved Software Vendor Program



Java Developer's Journal announced its first annual "Readers' Choice Awards" program three years ago as a patented and copyrighted innovation. *JDJ's* Readers' Choice Award program became to be the first true and openly tabulated and monitored media award program of its kind ever. As it happens in every new innovation, since then, we have seen a number of other media organizations and magazines imitate *JDJ's* Readers' Choice Awards.

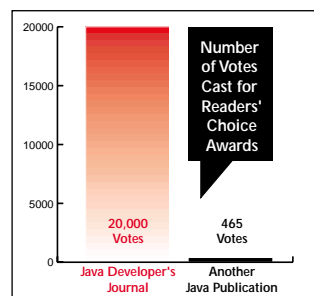
I would like to point out the glaring differences in both participation and the methodology used in *JDJ* and all other award programs. *JDJ* is the innovator and the only true implementer of the first open media program of this kind. This year as we announce and congratulate 17 winners and 51 finalists of *JDJ* Readers Choice Awards, we would like to remind you the reasons why *JDJ* Awards are also referred to as "the Oscars of the software industry." *JDJ's* Award Program is the first and only media industry awards tabulated and audited by BPA international, the leading magazine industry audit firm. They are given to superior products nominated and selected by a large participation of professional Java developers and *JDJ* readers. This year, over 20,000 *JDJ* readers actively voted to help us select these winning products. Also this year two other competing magazines introduced their first award programs. The first one of the other two magazines generated 468 total votes (according to their records) and the second magazine less than 1,000. If you divide 468 total votes to 10 products categories, 46 random people (most likely the employees of the participating companies) are deciding which products are the best of the year and these magazines are announcing these results to the world as the winners of their "Readers' Choice Awards."

So what does it really mean to you and to the software industry when you next time see an award winner logo next to a product ad? It may or may not mean much depending on what award they are referring to.

Introducing... the *JDJ* Approved Product/Vendor Program

We will keep our readers posted about future award programs and keep a close watch to these kind of programs. In the mean time I would like to proudly announce a new service for our readers; namely "*JDJ* Approved Product/Vendor" program. As a *JDJ* reader, you may already know that *Java Developer's Journal's* BPA audited circulation is higher than all other Java specific publications combined. You may have also realized that *JDJ's* average page count is twice, some times three times larger than our competitors'. We do heavily rely on our sponsors as we serve the Java software industry. However, one thing you may not have noticed is that we do keep our right to decline accepting advertisements from the vendors whose products we do not feel comfortable to endorse and we simply refuse to accept advertisement and sponsorships from certain companies. All our advertisers become our sponsors by invitation only and after a thorough check on the company. To serve our readers, such as yourself, who are among the decision makers in charge of software and product purchases, we recently announced the "*JDJ* Approved Product/Vendor Program."

Please visit JavaDevelopersJournal.com web site to find a list of vendors who we endorse as a true indication of superior Java products. This will help you to clear many questions while trying to identify a reliable product which will do the job for your small or large project and a dependable vendor behind the product. ☺



fuat@sys-con.com

Java Developer's Journal Readers' Choice Awards, the 'Oscars of the Software Industry,' Will Recognize 17 Winners and 51 Finalists During an Awards Ceremony at

 in Santa Clara, CA

As the world's leading publication targeting Java professionals, *Java Developer's Journal* announced at JavaOne the results of its annual awards recognizing the best software products providing business solutions with Java.

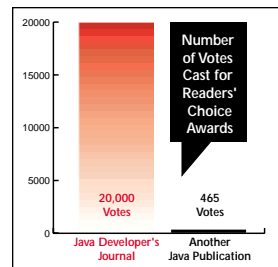
JDJ will present this year's Readers' Choice Awards during a ceremony to be held at JavaCon 2000 in September (see www.Java-Con2000.com for full details of location and program). The awards will be presented on September 25 to the winners and finalists before an audience at the **SYS-CON Radio** booth from which *JavaDevelopers-Journal.com* and **SYS-CON Radio** will be bringing this event live to the software industry and the readers of *JDJ*.

Over 20,000 *JDJ* Readers Vote for the 68 Best Java Products

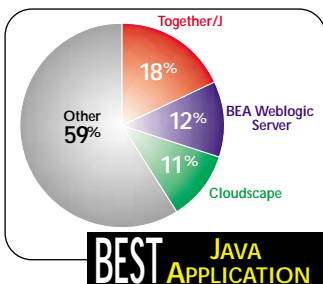
This year more than 20,000 *JDJ* readers cast their votes (compared to a 400-vote turnout by another Java publication's award program imitating that of *JDJ*) to select the best products of the year. The polls

opened on the **SYS-CON.com** web site on January 17, 2000, and ran through May 31, 2000. The votes have been tabulated and audited for accuracy by BPA International, the largest international audit firm specializing in magazine industry circulation audits.

At the JavaCon 2000 presentation ceremony winners and finalists in 17 award categories will be acknowledged for the contributions they have made in developing Java-based solutions that respond to and meet the increasing demands of the industry.



Awards are given as a result of **SYS-CON Media's** unique and patented voting procedure, which is openly tabulated and audited for accuracy by an independent audit firm. The results of the vote distribution could closely simulate the market share and popularity of each product as a result of wide participation by industry professionals who are among *JDJ* readers.



— Winner —
TOGETHER/J
from TogetherSoft LLC
(aka Together Solo)

Together Solo is the multiplatform control center that delivers increased productivity to e-solution developers. Release 4.0 includes the functionality of Together/J and Together/C++ in one product. More than just modeling software, Together Solo provides a comprehensive, enterprise-wide backbone for development solutions in "Internet time." Together Solo delivers highly productive developers a full and complete range of must-have features, including



OneSource simultaneous round-trip engineering, true multiuser team support, reusable integrated patterns and components, multi-level documentation generation, customizable QA tracking for requirements, metrics and audits (now including bar charts and graphs), a new project expert and extensive customizability.

www.togethersoft.com

— 1st Finalist —
BEA WEBLOGIC SERVER
from BEA

BEA WebLogic Server is the leading Java application server and the de facto standard for quickly building and deploying reliable and sophisticated Web and e-commerce applications using the latest J2EE specifications, including Enterprise JavaBeans (EJB) components and Java Message Service (JMS). It supports significant new e-commerce technologies, including WML and XML, for the rapid development of mobile applica-



tions and business-to-business market places. **BEA WebLogic Server** delivers fast time-to-market with the scalability, performance and fault tolerance required for mission-critical e-commerce applications.

www.bea.com

— 2nd Finalist —
CLOUDSCAPE
from Cloudscape

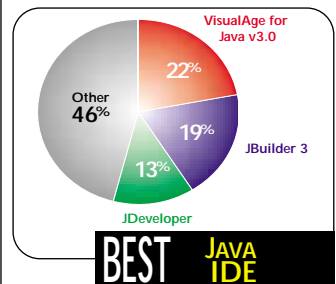
Cloudscape is a Java technology-based database for embeddable systems and distributed e-business applications. The recently upgraded **Cloudscape 3.5** consists of the **Cloudscape** database management system **Cloudsync** for data and application synchronization and **Cloudconnector**, a server framework for Internet connections to the **Cloudscape** DBMS. Optimized to work with Java applications, **Cloudscape 3.5** offers higher performance, improved reliability and greater multiuser support – from 100 to 250 concurrent users – as well as key new features critical to today's distributed applica-



tion environments, including database encryption and signed Jar files.

www.informix.com

— 3rd Finalist —
YOUCENTRIC
from YOUcentric, Inc.

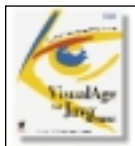


— Winner —
VISUALAGE FOR JAVA v.3.0
from IBM

IBM VisualAge for Java is developed using four key principles:

- Extend existing applications to e-business.
- Scale with high performance data solutions.

- Manage the development of large teams and projects.
- Build a development community by providing "how to" information.



It is a great honor to have our principles validated by our users, resulting in the three *JDJ* awards. We extend a welcome to all Java developers to become a member of our growing developer community. ☛

www.ibm.com/vadd

— 1st Finalist —
JBUILDER 3

from Inprise

JBuilder Enterprise is the most comprehensive set of award-winning visual development tools for creating pure Java enterprise-scale applications for the Java 2 platform. Rapidly develop and deploy e-business applications with VisiBroker for Java and the Inprise Application



Server 4.0, which provides complete support for the J2EE technology standard integrated with the proven scalability and high performance of CORBA. Deliver your distributed Java 2 applications using BeansExpress for EJBs and the Java 2 remote debugger. Also deliver mobile and disconnected computing applications using the Borland JDataStore Pure Java embeddable database with transactional and query support. ☛

www.inprise.com

— 2nd Finalist —
JDEVELOPER

from Oracle

Oracle JDeveloper is a full-featured application development tool that offers integrated support for building end-to-end e-business applications for the Internet. Oracle JDeveloper offers an integrated environment for application developers to productively develop, debug, deploy, reuse and customize multitier, component-based **ORACLE** Java and XML applications.

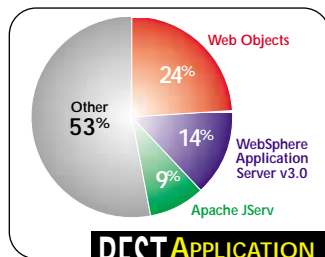
Oracle Business Components for Java is a standards-based, server-side framework for creating scalable, high-performance Inter-

net applications. The framework provides design-time facilities and runtime services to drastically simplify the task of building, debugging, customizing and reusing business logic. This component framework is available as an integral part of Oracle JDeveloper. ☛

<http://otn.oracle.com/products/jdev/> and www.oracle.com/java/

— 3rd Finalist —
VISUALCAFÉ

from Symantec



BEST APPLICATION SERVER

— Winner —
WEBOBJECTS

from Apple

WebObjects 4.5 is an open and flexible, standards-compliant application server that's ideal for developing Java applications. It has an object-oriented architecture that promotes quick development of reusable components. WebObjects provides scalability, load balancing and fault-tolerant capabilities for multithreaded applications running across multiple CPUs that support high transaction volumes. ☛



www.apple.com

— 1st Finalist —
WEBSHERE APPLICATION SERVER ENTERPRISE EDITION V.3.0

from IBM

IBM WebSphere Application Server Enterprise Edition enables full e-business transactions over the Web. Using open, standards-based technologies like interoperable Common Object Request Broker Architecture (CORBA) and Enterprise JavaBeans (EJB), Enterprise Edi-



tion provides comprehensive, high quality middleware runtime services for distributed component applications. ☛

www.ibm.com

— 2nd Finalist —
APACHE JSERV

from the Java Apache Project

Apache JServ is a 100% Pure Java servlet engine fully compliant with the JavaSoft Java Servlet APIs 2.0 specification. JServ is the result of the open volunteer project known as the Java Apache Project, which believes in the great value of complete portability between different servlet platforms, together with portability of Java binary code.



The Project worked very closely with JavaSoft and other JVM implementers to provide correct interpretations or Java specifications both for servlets and for the Java language itself and the result is a pure servlet engine that works on any "version 1.1 compliant" Java Virtual Machine and may execute any Java servlet compliant with version 2.0.

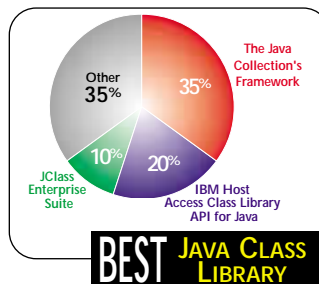
To obtain such complete abstraction from native environments, Apache JServ 1.0 was designed as a server application serving requests made with a specific protocol (simply enough named Apache JServ Protocol). Then, included modules allow the Web server to "translate" servlet requests and forward them to the right servlet engine.

At this moment the Apache JServ distribution contains only a module for the Apache Web Server (mod_jserv). ☛

<http://java.apache.org>

— 3rd Finalist —
WEBLOGIC SERVER

from BEA Systems Benelux



BEST JAVA CLASS LIBRARY

— Winner —
THE JAVA COLLECTIONS

from Sun Microsystems

The Java Collections Framework, a part of the Java 2 platform, is a unified architecture for representing and manipulating collections of objects. The Framework allows collections to be manipulated independently and reduces programming effort while increasing performance. It allows for interoperability among unrelated APIs, reduces effort in designing and learning new APIs and fosters software reuse. ☛



www.sun.com

— 1st Finalist —
IBM HOST ACCESS CLASS LIBRARY API FOR JAVA

from IBM

IBM Host Access Class Library (HACL) is an object-oriented API, providing significant improvement over traditional emulator programming interfaces like EHLLAPI. With HACL, application developers can write Java applets that manipulate data from the data stream presentation space (such as 3270, 5250, and Virtual Terminal) without requiring the users to have the applets residing on their machines. HACL Java applets can open a session to the host, wait for incoming host data, get specific strings from the imaginary screen and wait for the next host response plus many other host interactions. The entire operation can be done without ever showing host screens. ☛



www.ibm.com

— 2nd Finalist —
JCLASS ENTERPRISE

from KL Group

JClass Enterprise Suite is the world's most comprehensive collection of Java components. Fully scalable to mission-critical development environments, this award-winning suite provides a wide range of high-value GUI functionality including charting and graphing, tables and grids, data connectivity, data input and validation, JAR optimization, GUI enhancements, and layout and reporting –

YouCentric

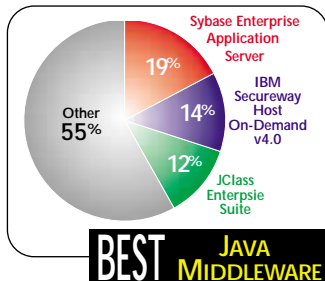
www.youcentric.com/nobrainier

JCLASS Enterprise Suite

everything the professional Java developer needs to build powerful and flexible application front ends. JClass components also offer IDE, JDK and platform support and include one year of Gold Support with free upgrades. ☛

www.klgroup.com

— 3rd Finalist —
JGL
from ObjectSpace



— Winner —
SYBASE ENTERPRISE APPLICATION SERVER
from Sybase

The highly scalable, robust application server for e-portal and Internet business solutions, **Sybase EAServer** provides industrial-strength services for deploying Web and distributed applications using core Java 2 Enterprise Edition (J2EE) standards, including Enterprise JavaBeans (EJB). EAServer offers cross-client and cross-component support for applications based on CORBA, XML, HTML, DHTML, ActiveX clients, PowerBuilder, COM, C and C++. It also provides a single point of integration for heterogeneous back-office systems and extends customer businesses to the Web. EAServer includes transaction and security management, load balancing, failover and high availability – enabling customers to support highly trafficked Web sites. ☛



www.sybase.com

— 1st Finalist —
WEBSHERE HOST ON-DEMAND v4
from IBM

Get secure Web-to-host access and e-business application programming support with **IBM WebSphere Host On-Demand**. One interface to all your TN3270E, TN5250, Virtual Terminal (VT) 50/100/220 and CICS applications, consistent across platforms. Maximize your productivity with standard desktop utilities, file transfer and host print. Create custom e-business applications using the Host Access Class Library, Host Access Beans for Java and Host Access ActiveX controls. A graphical user interface offers an alternative to host green screens. Web server installation means centralized deployment and administration, and users always access the latest version. Host On-Demand requires no client installation or middle-tier server, and it supports your choice of client and server platforms. ☛

www.ibm.com

— 2nd Finalist —
PROGRESS SONICMQ
from Progress

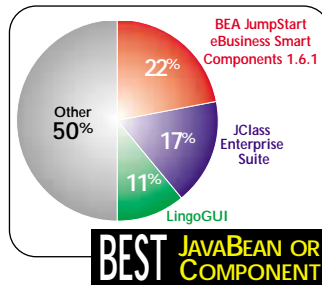
Progress SonicMQ is a fast, flexible, scalable Internet-resilient messaging server that simplifies the integration of distributed enterprise applications and building of e-commerce business solutions.

SonicMQ offers robust communications functionality, allowing applications to reliably handle disconnected or mobile users, network outages and software/hardware failures.

Since it began shipping last year, SonicMQ has won rapid acceptance among leading e-commerce infrastructure companies, among them GemStone Systems, IONA Technologies, IVC/BuildNet and SAGA Software. Other awards include World Class Award, *Java Developer's Journal*, and awards for best Java middleware, and best Java messaging tool. ☛

www.sonicmq.com

— 3rd Finalist —
VISIBROKER
from Borland Inprise



— Winner —
BEA JUMPSTART eBUSINESS SMART COMPONENTS

from BEA
(aka BEA WebLogic Commerce Server)

The award-winning **BEA WebLogic Commerce Server 2.0** allows you to build applications using an EJB-based set of server functions and BEA WebLogic Server's robust, proven architecture. BEA's commerce server enables fast deployment of personalized e-commerce applications that can be modified instantaneously to meet customer demands or take advantage of new market opportunities. The BEA WebLogic Commerce Server 2.0 provides a comprehensive set of prebuilt commerce functions, including a product catalog, shopping cart, order entry, fulfillment and customer-care functions. You can extend and customize the behav-



ior of more than 80 business components provided by BEA WebLogic Commerce Server. These components interact with each other and with J2EE services provided by the BEA WebLogic Server. ☛

www.bea.com

— 1st Finalist —
JCLASS ENTERPRISE SUITE
from KL Group

JClass Enterprise Suite is the world's most comprehensive collection of Java components. Fully scalable to mission-critical development environments, this award-winning suite provides a wide range of high-value GUI function-

JCLASS Enterprise Suite

ality including charting and graphing, tables and grids, data connectivity, data input and validation, JAR optimization, GUI enhancements, and layout and reporting – everything the professional Java developer needs to build powerful and flexible application front ends. JClass components also offer IDE, JDK and platform support and include one year of Gold Support with free upgrades. ☛

www.klgroup.com

— 2nd Finalist —
LINGOGUI
from SlangSoft
(renamed LingoBox)

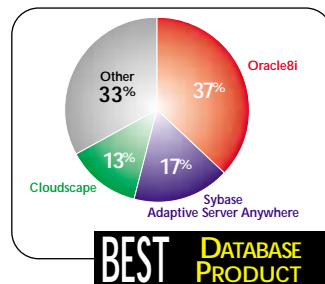
SlangSoft's **LingoGUI** (renamed **LingoBox**) is a small-footprint Java component (or applet) that adds smart, fast text input in 42 languages to the UI of any Web-based application, offering end users cross-platform text input in any national language without significant changes to the server-side architecture.

LingoBox can easily be added to any Web-based UI, maintaining the look and feel of normal HTML Web browser text components. The Lingo Box applet dynamically receives text input modules, rendering engine and fonts, linguistic dictionaries and virtual keyboards for up to 42 national languages via the Web in real time, requiring no installation, download or plug-in. A growing number of developers are using this cool Java component. ☛



www.slangsoft.com

— 3rd Finalist —
STUDIOJ
from Rogue Wave Software



WebVision

www.webvision.com

— Winner —
ORACLE 8i
from Oracle

Oracle8i – the database for Internet computing – provides the industry's most complete and comprehensive platform for building and deploying Internet and enterprise applications. Oracle8i, the most popular database in the world, is already used by most top global Internet brands such as Amazon.com, CDNow and Excite, and many emerging companies. Oracle8i is revolutionizing the way companies better serve and retain customers, expand their markets and streamline processes.

For the Java developer, Oracle8i offers a complete Java 2 (JDK 1.2 API level) environment right inside the database enabling Java classes, Java database, stored procedures, servlets, JSPs and EJBs to benefit from the high performance, availability and security of the Oracle8i database.

www.oracle.com/java/

— 1st Finalist —
SYBASE ADAPTIVE SERVER ANYWHERE
from Sybase

Adaptive Server Anywhere, the leading workplace DBMS, offers an easy-to-use, affordable and scalable SQL database for workgroups of any size. Simplified installation and administration, a small footprint, transaction processing and extensive SQL capabilities make Adaptive Server Anywhere ideal for widespread deployment in workgroup, mobile, or embedded environments. Features include a combined stored procedure and Java debugger, an easy connection across firewalls, an easy unload and reload, and fast table truncations. Adaptive Server Anywhere also includes a stored procedure debugger, Euro support and additional Java in the database functionality – in particular, multithreading of Java classes inside the database.

www.sybase.com

— 2nd Finalist —
CLOUDSCAPE
from Cloudscape

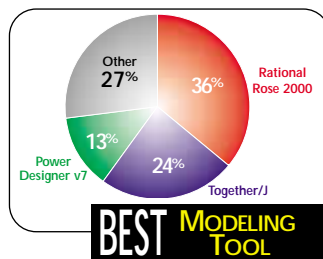
Cloudscape is a Java technology-

based database for embeddable systems and distributed e-business applications.

The recently upgraded Cloudscape 3.5 consists of the Cloudscape database management system Cloudsync for data and application synchronization and Cloudconnector, a server framework for Internet connections to the Cloudscape DBMS. Optimized to work with Java applications, Cloudscape 3.5 offers higher performance, improved reliability and greater multiuser support – from 100 to 250 concurrent users – as well as key new features critical to today's distributed application environments, including database encryption and signed Jar files.

www.informix.com

— 3rd Finalist —
JDATASTORE
from Borland Inprise



— Winner —
RATIONAL ROSE
from Rational Software

Rational Rose provides Unified Modeling Language (UML)-based modeling for designing component-based applications. The UML, pioneered by Rational and officially adopted as a standard by the Object Management Group (OMG), is the industry-standard language for specifying, visualizing, constructing and documenting the artifacts of a software system. Rational Rose features multilanguage capabilities and enterprise team development features, including integration with Rational's ClearCase software configuration management product. Multiplatform Java development round-trip engineering is based on JDK 1.2, with support for pure Java, Java Bean property generation, nested classes and components such as jars, zips and cabs.

www.rational.com

— 1st Finalist —
TOGETHER/J
from TogetherSoft LLC
(aka Together Solo)

Together Solo is the multiplatform control center that delivers increased productivity to e-solution developers. Release 4.0 includes the functionality of Together/J and Together/C++ in one product. More than just modeling software, Together Solo provides a comprehensive, enterprise-wide backbone for developing solutions in



“Internet time” Together Solo delivers highly productive developers a full and complete range of must-have features, including OneSource simultaneous round-trip engineering, true multiuser team support, reusable integrated patterns and components, multi-level documentation generation, customizable QA tracking for requirements, metrics and audits (now including bar charts and graphs), a new project expert and extensive customizability.

www.togetherso.com

— 2nd Finalist —
POWER DESIGNER v7
from Sybase

PowerDesigner is the next-generation modeling tool that combines the functionality of Sybase's leading database design product with powerful UML-based object modeling to create the first totally integrated object/relational analysis and design environment. PowerDesigner solves the mismatch involved in persisting object-oriented applications in a relational database. PowerDesigner also helps you take the first steps in creating distributed applications using UML-based class diagrams to model your application objects and then automatically generate PowerBuilder nonvisual objects (NVO) and Java.



www.sybase.com

— 3rd Finalist —
COOL:JOE
from Sterling Software



— Winner —
JAVA IN A NUTSHELL
by David Flanagan
O'Reilly & Associates

Java in a Nutshell contains an advanced introduction to Java and its key APIs and provides quick reference material on all the classes and interfaces in the following APIs: java.lang, java.io, java.beans, java.math, java.net, java.security, java.text, java.util and javax.crypto. (668 pages, \$29.95)



— 1st Finalist —
JAVA SERVLET PROGRAMMING
by Jason Hunter
O'Reilly & Associates

Java Servlet Programming covers how to write effective servlets. Topics include serving dynamic Web content, maintaining state information, session tracking, database connectivity using JDBC, and applet-servlet communication. (528 pages, \$36.95)



— 2nd Finalist —
ENTERPRISE JAVABEANS
by Richard Monson-Haefel
O'Reilly & Associates

Enterprise JavaBeans is a thorough introduction to EJB 1.1 and 1.0 for the enterprise software developer. It shows you how to develop enterprise beans to model your business objects and processes. The EJB architecture provides a highly flexible system in which components can easily be reused, and which can be changed to suit your needs without upsetting other parts of the system.

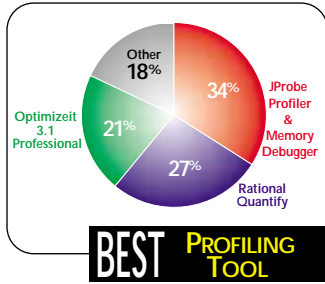


StarBase

www.starbase.com

Enterprise JavaBeans teaches you how to take advantage of the flexibility and simplicity that this powerful new architecture provides. (492 pages, \$34.95) 🍌

— 3rd Finalist —
JAVA 2D GRAPHICS
 by Jonathan Knudsen



— Winner —
JPROBE PROFILER & MEMORY DEBUGGER
 from KL Group

JProbe Profiler helps you quickly eliminate performance bottlenecks caused by inefficient algorithms in your Java code. JProbe Profiler combines a visual call graph interface, a unique garbage monitor and advanced data collection technology to provide you with highly accurate performance diagnostics, including line-by-line results. JProbe Memory Debugger (fully integrated with JProbe Profiler) pinpoints the cause of memory



leaks in Java applications by tracking which objects hold references to other objects. Visualize memory usage, including memory allocation and garbage collection, in real time with the intuitive Memory Usage window. 🍌

www.klgroup.com

— 1st Finalist —
RATIONAL QUANTIFY
 from Rational Software

Rational Quantify is an advanced

performance profiling tool that automatically and quickly pinpoints application performance bottlenecks, taking the difficulty and guesswork out of performance tuning. Quantify for Windows NT also delivers repeatable timing data for all parts of an application, including components, not just the parts for which you have the source code.

Graphical performance data views allow developers to see exactly how their program executed. Viewing the code in terms of function call architecture as it is executed gives developers the insight they need to make the right changes in the program to improve performance. Detailed tabular views and line-by-line timing data take developers right to the source of bottlenecks. With Quantify for Windows NT, **Rational** no development time will be wasted tuning code that does not offer significant returns in performance. 🍌

www.rational.com

— 2nd Finalist —
OPTIMIZEIT 3.1 PROFESSIONAL
 from Intuitive Systems, Inc.

Optimizeit is a powerful profiling solution for Java developers looking to rapidly track down and fix performance issues such as memory leaks and performance bottlenecks in any Java program. This award-winning Java profiler offers a complete toolkit including memory leak debugging, CPU profiling and monitoring of object allocations.

Optimizeit provides essential profiling of application server environments, enhanced by instant integration with most popular application servers, unique overhead limiting features ensuring scalability, and new offline profiling to allow testing of applications in production environments.

Highly flexible, Optimizeit allows local and remote profiling of Java processes running on Windows, Sparc Solaris, Solaris Intel and Linux, and supports most JDK



1.1x and Java 2 virtual machines with no modifications. 🍌

www.optimizeit.com

— 3rd Finalist —
TRUETIME JAVA EDITOR
 from Compuware NuMega



— Winner —
STYLE REPORT
 from InetSoft Technology Corporation

Style Report is a family of advanced printing/reporting tools for Java programmers. Style Report/Pro provides a powerful layout engine for complex reports and serves as the foundation of the product family. It has numerous built-in features such as charting, grouping/summarization and PDF generation. Style Report/Pro's flexible, easy-to-use API and



visual tools put all the power directly into developers' hands. Style Report/DataSource is built from a modern, multitier-enabled query engine that makes reporting from XML, EJB, JDBC/SQL and CORBA a simple visual exercise. Style Report/Enterprise publishes complex reports interactively on the Internet/intranet for users who only need a simple Web browser. 🍌

www.inetsoft.com

— 1st Finalist —
DBPROBE 4.11
 from InterNetivity Inc.

Business Intelligence that's easy to deploy, easy to use and scalable to thousands of users: **InterNetivity dbProbe 4.1** delivers the

sweet spot in Web-based decision support. Its powerful, fully interactive OLAP and standard reporting client allows Web-based users to easily access and analyze data from a variety of sources. Data sources include ODBC and Microsoft OLE DB



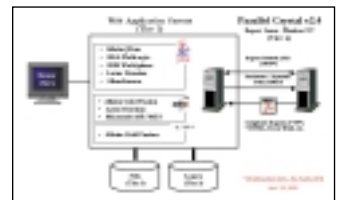
for OLAP-compliant servers. Written in 100% Pure Java, the dbProbe client lets users drill down, slice and dice, graph, create and share reports – and more. The dbProbe client occupies just 410 kilobytes, supports mobile users and offers fast performance. 🍌

www.internetivity.com

— 2nd Finalist —
PARALLEL CRYSTAL REPORT SERVER

from Dynalivery Corporation formerly

Mobile Application Servers Inc. **Parallel Crystal** is a multiuser, enhanced version of Seagate Software's popular Crystal Reports report generating software. It is a software development kit (SDK) for integrating leading Web application servers with Crystal Reports.



The key features of Parallel Crystal are:

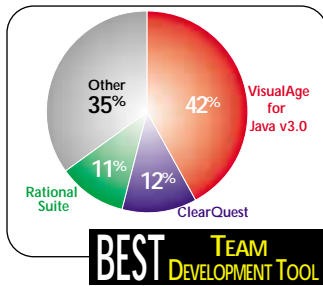
1. Capability to integrate with the major Java and COM-based Web application servers, including sample code and professional technical support services
2. High fidelity output using Adobe Portable Document Format (PDF)
3. High scalability and report server uptime
4. Compatibility with report design templates created with Seagate Crystal Reports 🍌

www.dynalivery.com/parallelcrystal.htm

— 3rd Finalist —
JCLASS PAGELAYOUT
 from KL Group

NetDive

www.netdive.com



— Winner —
VISUALAGE FOR JAVA v3.0
 from IBM

- IBM VisualAge for Java is developed using four key principles:
- Extend existing applications to e-business.
 - Scale with high-performance data solutions.
 - Manage the development of large teams and projects.
 - Build a development community by providing "how to" information.

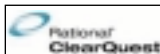


It's a great honor to have our principles validated by our users, resulting in the three *JDJ* awards. We extend a welcome to all Java developers to become a member of our

growing developer community. www.ibm.com/vadd

— 1st Finalist —
CLEARQUEST
 from Rational Software

Rational ClearQuest is a highly flexible defect and change-tracking system that captures and tracks all types of change for any type of project. The fully customizable interface and workflow engine enable ClearQuest to adapt to any development process. With support for industry standard databases, ClearQuest scales to support projects of any size. Integration with other development solutions, including configuration management, automated testing and requirements management tools, ensures that all members of your team are tied into the defect and change-tracking process.



www.rational.com

— 2nd Finalist —
RATIONAL SUITE
 from Rational Software

Rational Suite is a multiplatform

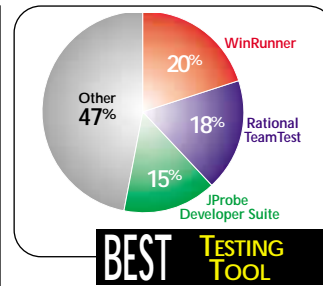
(Windows and UNIX) product family that speeds software development in the Internet era. Rational Suite products enable teams to define and manage requirements; architect and develop software; test system reliability, functionality and performance; and organize an effective development process.

- Specific editions of Rational Suite include Rational Suite AnalystStudio, Rational Suite DevelopmentStudio, Rational Suite DevelopmentStudio Real-Time Edition, Rational Suite TestStudio, Rational Suite PerformanceStudio and Rational Suite Enterprise. The Rational Suite product family integrates seamlessly with Rational ClearCase, Rational's market-leading configuration management solution.



www.rational.com

— 3rd Finalist —
STARTEAM 4.1
 from StarBase



— Winner —
WINRUNNER
 from Mercury Interactive

WinRunner is a functional testing tool that verifies applications work as expected. By capturing, verifying and replaying user interactions automatically, WinRunner identifies defects and ensures that business processes that span across multiple applications and databases work flawlessly the first time and remain reliable throughout the life cycle. WinRunner supports the most extensive range of Java environments, including industry-leading virtual machines such as



QuickStream

www.quickstream.com

Internet Explorer, Netscape, Sun's Java Plug-in and Applet Viewer, Oracle JInitiator, JDK/JRE, and Microsoft's Jview and Java toolkits including Java Foundation Classes, Symantec VisualCafé, KL Group JClass, Oracle Developer and Sun's AWT, among others. ☉

www.merc-int.com

— 1st Finalist —
**RATIONAL
 TEAMTEST**

from Rational Software

Rational TeamTest is a market-leading functional testing tool, delivering seamlessly integrated functional testing of Web, ERP and client/server applications. Built on a scalable, integrated server-based test repository, Rational TeamTest combines leading-edge testing power and comprehensive management tools to set the standard for automated testing of Web, ERP and client/server applications. Rational TeamTest comprises the following component products: Rational Robot (test recording tool), Rational Clear-



Quest / TT Edition (defect tracking), Rational SiteCheck (Web site management), TestManager WebEntry (Web-based defect entry tool) and TestManager (test planning, management and analysis tool). ☉

www.rational.com

— 2nd Finalist —
**JPROBE
 DEVELOPER SUITE**
 from KL Group

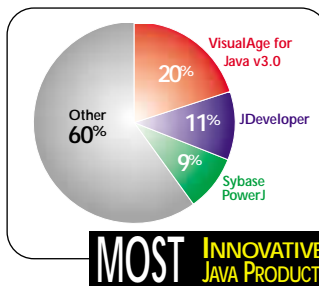
JProbe Developer Suite provides powerful performance profiling, memory debugging, code coverage and thread analysis capabilities in one conveniently integrated suite designed for client-side Java development. With JProbe Developer Suite in your tool belt, tuning your code has never been easier. The JProbe products help you to understand precisely what's causing problems and where, right down to the offending line of source code. JProbe Developer Suite paints an intuitive, graphical picture of everything from memory usage to calling relationships, helping you navigate your way to the



root of the problem quickly and easily. ☉

www.klgroup.com

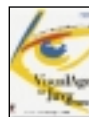
— 3rd Finalist —
**NUMEGA
 DEVPARTNER
 JAVA EDITOR**
 from Compuware Corporation



— Winner —
**VISUALAGE
 FOR JAVA v3.0**
 from IBM

VisualAge for Java is developed using four key principles:

- Extend existing applications to e-business.



- Scale with high-performance data solutions.
- Manage the development of large teams and projects.
- Build a development community by providing "how to" information.

It's a great honor to have our principles validated by our users, resulting in the three **JDJ** awards. We extend a welcome to all Java developers to become a member of our growing developer community. ☉

www.ibm.com/vadd

— 1st Finalist —
JDEVELOPER
 from Oracle

Oracle JDeveloper is a full-featured application development tool that offers integrated support for building end-to-end e-business applications for the Internet. Oracle JDeveloper offers an integrated environment for application developers to productively develop, debug, deploy, reuse and customize multitier, component-based Java and XML applications.



Syntion AG

www.syntion.com

Oracle Business Components for Java is a standards-based, server-side framework for creating scalable, high-performance Internet applications. The framework provides design-time facilities and runtime services to drastically simplify the task of building, debugging, customizing, and reusing business logic. This component framework is available as an integral part of Oracle JDeveloper. ☪

www.oracle.com/java/

— 2nd Finalist —
POWERJ

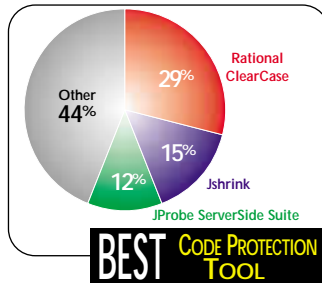
from Sybase

PowerJ provides a true end-to-end solution for building sophisticated Internet applications, exploiting the benefits of HTML, Java clients, and delivering powerful Java server-side components. PowerJ not only offers powerful database capabilities, it also integrates seamlessly with Sybase Enterprise Application Server, enabling enterprise-class applications from creation to testing and debugging to deployment. PowerJ and EAServer

exploit the benefits of HTML, Java clients. This combination delivers powerful Java server-side components. PowerJ also leverages skill sets, applications and data with the support of PowerBuilder DataWindows, multiple component models and virtually any database. ☪

www.sybase.com

— 3rd Finalist —
TOGETHER/J
from Togethersoft



— Winner —
CLEARCASE

from Rational Software

Rational ClearCase simplifies the process of change via comprehensive software configuration management – including version control, workspace management, process control and build management – with a transparent, non-intrusive approach.

With Rational ClearCase, e-development teams can accelerate development cycles, ensure the accuracy of releases, reliably build and patch previously shipped products, and organize an automated development process – all without changing their environment or their tools.

Rational ClearCase 4.0 facilitates e-business development efforts by unifying content authors with the rest of the e-development team. Rational ClearCase 4.0 enables multiple content authors to work in parallel on XML and HTML content and automatically reconcile their changes. ☪

www.rational.com

— 1st Finalist —
JSHRINK

from Eastridge Technology

Jshrink removes unused code, data and symbolic names from compiled Java class files, resulting in smaller files that load faster and yield less information when decompiled. Unlike other obfuscators, Jshrink doesn't change public or protected fields and methods and can therefore be used on redistributable components such as class libraries. Size reductions above 30% are possible. Jshrink was first released in May 1997 and was the first commercial Java obfuscator on the market. Jshrink is available both as a native Windows application and in a Pure Java version. ☪



www.e-t.com

— 2nd Finalist —
**JPROBE
SERVERSIDE SUITE**
from KL Group

JProbe ServerSide Suite provides

InetSoft

www.inetsoftcorp.com

powerful performance profiling, memory debugging, code coverage and thread analysis capabilities in one conveniently integrated suite designed for server-side Java development. JProbe products help you to understand precisely what's causing problems in your code and where, right down to the offending line of source code.

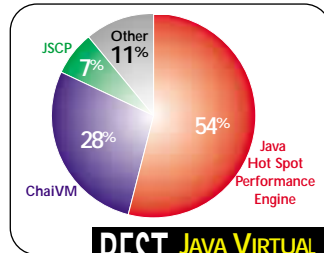
JProbe products paint an intuitive, graphical picture of everything from memory usage to calling relationships, helping you navigate your way to the root of the problem quickly and easily. ServerSide Editions of JProbe products include the Server Launch Pad,



enabling convenient integration with your favorite Web or application server. ☛

www.klggroup.com

— 3rd Finalist —
4TH PASS
SOURCEGUARD
 from 4th Pass



BEST JAVA VIRTUAL MACHINE

— Winner —
JAVA HOT SPOT PERFORMANCE ENGINE
 from Sun Microsystems, Inc.
 The Java HotSpot Server VM (previously known as the Java HotSpot Performance Engine) delivers the highest possible performance to

the server. It offers extremely reliable, secure and maintainable environments that meet the demands of today's enterprise customers. Version 2.0 of the Java HotSpot Server VM contains the Java HotSpot runtime environment and the server compiler and supports the Java 2 Platform, Standard Edition v.1.2.2 and v 1.3. This configuration delivers the highest possible performance and reliability to the server environment. It's currently available for download free of charge. ☛

<http://java.sun.com/products/hotspot/2.0/download.html>

— 1st Finalist —
CHAI VM
 from Hewlett Packard

ChaiVM is a complete environment for running Java programming language applications on information appliances. ChaiVM provides a smaller memory footprint and better performance and



scalability than competitive products while maintaining compatibility with the Java programming language and Java virtual-machine specifications.

Several notable features have been added into ChaiVM's efficient implementation with the latest release, 3.0, including applet support, reflection, object serialization, remote method invocation (RMI)1, as well as Zip and Jar2 archival and retrieval support. ☛

www.hewlett-packard.com

— 2nd Finalist —
JSCP
 from NSISCom

Based on authentic PersonalJava from Sun, JSCP products are designed to provide reduced time-to-market and shorter development cycles. By precisely controlling the resources used in Java applications – memory, CPU and input/output (I/O) channel – JSCP offers the highest reliability for Java imple-



ComputerWork.com

www.computerwork.com

Ajile

www.ajile.com

Zsigo

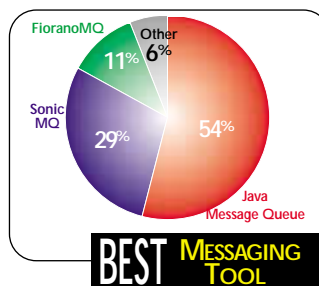
www.zsigo.com

mentation. Just-in-time (JIT)-based technology, optimized for the embedded environment, contributes to JSCP's high performance. Far more than just a productized JVM, JSCP offers a suite of tools and services for fast, reliable customizing of Java for specific customer hardware. Continuously updating to the latest version of the PersonalJava Compatibility Kit, NSIcom offers TCK certification service to its customers. ☛



www.nsicom.com

— 3rd Finalist —
JEODE PLATFORM
from Insignia Solutions



— Winner —
JAVA MESSAGE QUEUE

from Sun Microsystems, Inc. **Java Message Queue 1.0** software allows developers to focus on creating business logic rather than worrying about deployment topologies and data communication variables. Java Message Queue software is a full production implementation of the open-standard Java Message Service (JMS) 1.0.1 specification, which provides significant improvements over traditional process-oriented messaging systems. ☛



www.sun.com

— 1st Finalist —
SONICMQ
from Progress Software
Progress SonicMQ is a fast, flexi-

ble, scalable Internet-resilient messaging server which simplifies the integration of distributed enterprise applications and the building of e-commerce business solutions.

SonicMQ offers robust communications functionality, allowing applications to reliably handle disconnected or mobile users, network outages and software/hardware failures.

Since it began shipping last year, SonicMQ has won rapid acceptance among leading e-commerce infrastructure companies, among them GemStone Systems, IONA Technologies, IVC/BuildNet and SAGA Software. Other awards include World Class Award, *Java Developer's Journal*, and awards for best Java middleware, and best Java messaging tool. ☛

www.sonicmq.com

— 2nd Finalist —
FIORANOMQ
from Fiorano

FioranoMQ is the fastest, fully JMS-compliant Java messaging server for building high-performance, scalable and secure distributed enterprise applications. Only FioranoMQ uses a file-based message store to deliver guaranteed messages significantly faster than any other JMS implementation. In addition to supporting all the documented Publish/Subscribe and PTP (Point-to-Point) functionality in the JMS API, FioranoMQ adds a highly configurable security system, unmatched scalability, Web integration, superior XML interoperability, and bridging technologies to IBM MQSeries and MSMQ. ☛



www.fiorano.com

— 3rd Finalist —
IBUS
from Softwired

So What?

It seems that JavaOne isn't just Nerd Nirvana anymore...this is a good sign.



es, I was in San Francisco last month – but unlike the other 25,000 pilgrims, I wasn't fortunate enough to pay full homage to the Mecca of Java: JavaOne 2000. Instead, I was trapped in somewhat less than invigorating business meetings. I was also in a hurry to get back to Dallas to my wife and my brand new

(two-week old) baby boy. However, I did manage to sneak out for a few hours on two separate days to the Moscone Center in downtown San Francisco to sip from the Java technology cup....

Although my impressions of JavaOne this year are based mainly on this brief interlude, I've been following the conference closely via the Web and other means. And let me tell you one thing, folks – I was disappointed. Later I talked to several of my friends and fellow nerds who had attended the full conference. They concurred. This year the excitement and the energy were nowhere close to last year's JavaOne.

Later, on my Dallas-bound plane, my thoughts wandered back to JavaOne. And suddenly the reason why this year's show seemed more lukewarm hit me: there were very few surprises! Think about it. What was new at this conference? Granted, there were 150 technical sessions. All the hotels were booked as usual. Gimmicks abounded, congregations of nerds were found aplenty and there was much pomp and show. So what? As far as the Java platform itself is concerned, everything was old news. EJB architecture? Announced last year. The three lives of the Java Platform – J2ME/J2SE/J2EE? Announced last year. Jini, Java 2D, Collections, HotSpot – all these are last year's news. The small device market? Well, 3Com and Sun had much to say about it – last year. Microsoft's JVM? Last year's controversy.

Version 1.3 of the JDK was released a couple of months before JavaOne this year. But this release didn't introduce any new APIs. In contrast, last year's announcement of Java 2 was replete with new APIs, a redefined vision of Sun's Java Platform and major enhancements in the Java language. The 1.3 release is mainly bug fixes, performance improvements and enhancements to some of the existing APIs. No groundbreaking announcements this year.

At JavaOne 2000 Sun made numerous announcements including major initiatives with the Java Community Process, Java Web Start software and Forté for Java, Community Edition. Sun also updated the industry on initiatives including XML, JavaServer Pages technology, the Enterprise JavaBeans architecture and so on. However, all these are developments related to existing initiatives. Most of these are not technology advancements but, rather, tactical and strategic decisions for the Java community.

Application Server Focus

One thing was clear. J2EE has definitely come of age. The app server offerings now extend beyond the EJB object model. Vendors are bundling tools for application integration, XML support, workflow management and a plethora of other framework components that go into creating enterprise-level distributed applications. The app server

now occupies such a prominent space in business application development that **JDJ** is dedicating this issue to that market. A magazine focused exclusively on this subject is also under consideration; what do you, our readers, think of this initiative?

No More Nerd Nirvana?

Every year, JavaOne serves as the Nerd Nirvana for Java enthusiasts. Those of us fortunate enough to go to JavaOne spend a week immersed in Java and related technologies, away from the other mundane activities that are Simply Not Java. The nerd in us looks for cool technologies and the latest/greatest gadgets, and to interact with other similar birds that migrate to San Fran this time of the year. If this year's JavaOne is any indication, Java may be moving away from that. After all, how long can a technology continue to invent new areas for developers to dabble in?

Nevertheless, It's a Good Sign

Don't get me wrong. This lack of activity, while it may be a letdown for the developer community, is actually good. It signals the maturity of the Java Platform. For the past four years the Java Platform APIs and products have evolved rapidly and radically. Several 90-degree turns have taken place in Java technology. This year it seems the dust has settled. The products in the market are far more mature. The IDEs are already defined. The application server vendors have a much clearer definition of their offerings. J2EE is actually being deployed in the enterprise and real-world stories are completing the feedback loop for business applications. Java has made clear its role in the middle-tier and server-side architectures. EJB is gaining wider acceptance in the computing world. And software architects, designers and developers are concentrating on harder problems like persistence and distributed transactions using Java technologies.

Breaking Up May Be a Good Thing

I believe JavaOne is now too big a conference in terms of the number of participants and the ground it tries to cover both technologically and from a business perspective. Maybe it should be broken up into more focused sections. As a start, it could be divided into two conferences – one for developers, the other for business solutions. Perhaps these should be held in different geographical locations, such as JavaOne East Coast and JavaOne West Coast. It could even develop into a road show, like C++ World hosted by SIGS, and be held at different locations throughout the world.

SYS-CON Radio

One last thing. As it also did last year, **SYS-CON Radio** occupied a very prominent spot at JavaOne. The booth was filled with industry leaders who shared with attendees news of the various developments in Java technology. Hats off to the **JDJ** crew who made this a big success. The many interviews are available at the **JDJ** Web site, www.sys-con.com/java. ☪

ajit@sys-con.com

AUTHOR BIO

Ajit Sagar is editor-in-chief of XML-Journal and E-Commerce editor of JDJ. His regular E-Java column will return next month.

Career Opportunities

Career Opportunities

ADVERTISER INDEX

ADVERTISER	URL	PH	PG
4TH PASS	WWW.4THPASS.COM	877-484-7277	63
AJILE SYSTEMS	WWW.AJILE.COM	408-557-0829	100
ALLAIRE CORPORATION	WWW.ALLAIRE.COM/DOWNLOAD	888-939-2545	21
AMAZON.COM	WWW.AMAZON.COM/JAVA		37
APPEAL VIRTUAL MACHINES	WWW.JROCKIT.COM	468-402-2873	61
BEA	WWW.BEA.COM	800-817-4BEA	9
BULL	WWW.BULL.COM		17
CAPE CLEAR	WWW.CAPECLEAR.COM	353-1-241-9900	25
CAREER OPPORTUNITY ADVERTISERS		800-846-7591	102,118-133
CEREBELLUM SOFTWARE	WWW.CEREBELLUMSOFTWARE.COM	888-862-9898	35
CLOUDSCAPE	WWW.CLOUDSCAPE.COM	510-239-1900	12-13
COMPUTERWORK.COM	WWW.COMPUTERWORK.COM	800-691-8413	99
COMPUWARE	WWW.COMPUWARE.COM/NUMEGA	800-4-NUMEGA	19
ELIXIR TECHNOLOGY	WWW.ELIXIRTECH.COM/ELIXIRIDE	65 532-4300	57
FIORANO	WWW.FIORANO.COM	800-663-3621	65
FLASHLINE.COM, INC.	WWW.FLASHLINE.COM	800-259-1961	49
GEMSTONE	WWW.GEMSTONE.COM/WELCOME	503-533-3000	11
HOTDISPATCH.COM	WWW.HOTDISPATCH.COM	650-234-9752	27
IAM CONSULTING	WWW.IAMX.COM	212-580-2700	73
INETSOF TECHNOLOGY CORP	WWW.INETSOFTECH.COM	732-235-0137	55,98
INTUITIVE SYSTEMS, INC	WWW.OPTIMIZEIT.COM	408-245-8540	47
IONA TECHNOLOGIES	WWW.IONA.COM	800-672-4948	79
JAVACON2000	WWW.JAVACON2000.COM		84-85
JAVACON2000	WWW.JAVACON2000.COM		103-117
JDJ READERS' CHOICE AWARDS	WWW.SYS-CON.COM		86
JDJ STORE.COM	WWW.JDJSTORE.COM	888-303-JAVA	01
JV SEARCH	WWW.JVSEARCH.COM		20
KL GROUP INC	WWW.KLGROUP.COM/POWER	888-361-3264	15
KL GROUP INC	WWW.KLGROUP.COM/COURSE	888-361-3264	81
KL GROUP INC	WWW.KLGROUP.COM/GREAT	888-361-3264	136
MACROMEDIA	WWW.MACROMEDIA.COM/ULTRADEV	415-252-2000	32-33
METAMATA, INC.	WWW.METAMATA.COM	510-796-0915	77
NETDIVE.COM	WWW.NETDIVE.COM	415-981-4546	95
NEW ATLANTA	WWW.SERVLETEXEC.COM	800-GO-UNIFY	53
NO MAGIC	WWW.MAGICDRAW.COM	303-914-8074	7
NORTHWOODS SOFTWARE CORPORATION	WWW.NWOODS.COM	800-226-4662	70
OBJECTWAVE	WWW.OBJECTWAVE.COM	321-269-0111	70
OOP.COM	WWW.OOP.COM	877-667-6070	68-69
PARASOFT	WWW.PARASOFT.COM	888-305-0041	83
PERSISTENCE	WWW.PERSISTENCE.COM	650-372-3600	29
PRAMATI	WWW.PRAMATI.COM	877-667-PRAMATI	71
PROGRESS SOFTWARE	WWW.SONICMQ.COM/AD11.HTM	800-989-3773	2
PROTOVIEW	WWW.PROTOVIEW.COM	800-231-8588	3,135
QUEST SOFTWARE	WWW.QUEST.COM	949-754-8000	39
QUICKSTREAM SOFTWARE	WWW.QUICKSTREAM.COM	888-769-9898	96
SEGUE SOFTWARE	WWW.SEGUE.COM	800-287-1329	4
SIC CORPORATION	WWW.SIC21.COM	822.227.398801	75
SLANGSOFT	WWW.SLANGSOFT.COM	972-2-648-2424	43
SOFTWARE AG	WWW.SOFTWAREAG.COM/BOLERO	925-472-4900	67
STARBASE	WWW.STARBASE.COM	888-STAR700	93
SYBASE	WWW.SYBASE.COM/PRODUCTS/SERVER	800-8-SYBASE	45
SYNTION AG	WWW.SYNTION.COM		97
THINWEB TECHNOLOGIES	WWW.THINWEB.COM	877-THINWEB	51
TIDESTONE TECHNOLOGIES	WWW.TIDESTONE.COM	800-884-8665	23
TOGETHERSOFT CORPORATION	WWW.TOGETHERSOFT.COM	919-833-5550	6
UNIFY CORPORATION	WWW.EWAVECOMMERCE.COM	800-GO UNIFY	59
VISICOMP	WWW.VISICOMP.COM/JDJ7	831-335-1820	31
VISUALIZE	WWW.VISUALIZEINC.COM	602-861-0999	20
WEBVISION	WWW.WEBVISION.COM	800-531-5057	91
WIRELESS DEVELOPER	WWW.WIRELESSDEVELOPER.COM	800-594-5102	100
YOUCENTRIC	WWW.YOUCENTRIC.COM/NOBRAINER	888-462-6703	89
ZUCOTTO	WWW.ZUCOTTO.COM	613-789-0090	41

Master new skills at the largest gathering of Java Software Developers this fall!

(Plus, you'll learn directly from industry experts including the Java Developer's Journal editorial team!)

Register before August 25th and save \$200!



Building the New Enterprise

Conference:
September 24 – 27, 2000

Exhibition:
September 25– 26, 2000

Santa Clara Convention Center
Santa Clara, CA

Event Highlights

- Customize your program. Select from 5 tracks - over 60 sessions with 100+ hours of instruction.
- Special Pre-Conference Tutorial on Sunday – choose from 5 in-depth sessions.
- Night School Sessions – Extend your day program or take independently. Select from 15 in-depth topics on Sunday, Monday and Tuesday night.
- Get the latest insights and usable advice on the latest Java advancements from industry gurus and respected experts.
- Gold Passport allows you to take advantage of all sessions most economically.
- Two-day Interactive Exhibition on Monday and Tuesday.

www.JavaCon2000.com

Premier Corporate Sponsors:



Presented By:



Produced By:



Unify Launches eWave Engine 2.0

(San Jose, CA) – Unify Corporation has announced Unify eWave Engine 2.0, offering customers a feature-rich, highly scalable platform for rapid development of customized B2B and B2C applications.

Unify eWave Engine 2.0 incorporates the latest industry standards, and is the industry's only EJB 1.1 application server to provide dynamic replication capabilities for deploying and



redeploying EJB components around the clock and without

interruption to the Web site, allowing continuously available e-commerce. ☛

www.UnifyWave.com

Introducing SilverStream Application Server 3.5

(San Francisco, CA) – SilverStream Software, Inc., has announced a new version of its award-winning J2EE application server. The SilverStream Application Server 3.5 builds upon its ease-of-use capabilities by supporting best-of-breed Java IDEs, EJB development tools and JSP design tools from third parties. The new version also



provides a highly visual development and deployment environment for Linux desktops. ☛

www.silverstream.com

POET Releases OSS 6.1

(San Mateo, CA) – POET Software announced at the JavaOne Conference in San Francisco the release of its award-winning Object Server Suite (OSS) version 6.1, the essential object database for creating complex Java and C++ applications.



Key features include support for Java ODMG 3.0/JDK 1.2 Collections, tighter integration with leading Java IDEs, a new Java Enhancer, more efficient Forward Recovery Logging and the POET XML Export/Import Facility. ☛

www.poet.com

JRun 3.0 from Allaire in Marketplace

(Cambridge, MA) – Allaire Corporation has released JRun Server 3.0, delivering support for the

SYS-CON Media Announces Wireless Journal



Wireless Journal will be a mobile must-have. It will set the technology agenda, worldwide, for developers seeking to join or keep up with the fastest-growing area of software and hardware development since the

arrival of the microprocessor. From WAP and WML to unplugged advertising and non-voice services for the mass market, the coverage in *WJ* every issue will range from wireless soup to wireless nuts.

WJs premier issue is scheduled to be on North American newsstands in September.

Filled with savvy feature articles, up-to-the-minute product reviews, insightful industry commentary and colorful columns about unwired Life Beyond the PC,

Java 2 Platform, Enterprise Edition (J2EE) specification. Allaire also announced JRun 3.0 Studio, the industry's first integrated development environment for JavaServer Pages (JSP). ☛

www.allaire.com

Strategic Alliance for GemStone Systems, Techspan

(Beaverton, OR / Fairfax, VA) – GemStone Systems, Inc., and TechSpan, Inc., have announced a strategic alliance to provide integrated consulting services and support for e-businesses developing enterprise-class applications. The efforts of this partnership provide GemStone customers with the knowledge and technology required for faster integration and deployment of complex e-business sites. ☛



www.techspan.com

www.gemstone.com

TogetherSoft Releases Together Control Center 4.0

(San Francisco, CA) – At JavaOne TogetherSoft Corporation announced the official release of Together Control Center along with other parts of its expanded Together product line.

Together 4.0 features deeply integrated support for deploying applications to top-tier app servers from BEA, IBM and iPlanet; a new "EJB Assembler" diagram delivering a visual way to set up and maintain EJBs, permissions, and security roles; a new syntax-savvy editor with code completion; and an animated, distributed, multithreaded debugger fully integrated so developers can utilize the Together environment throughout the development process. The products are available for download from

www.togethersoft.com. ☛



Codagen Releases Gen-It for Java 1.2

(San Francisco, CA) – Codagen Technologies Corp. has come up with a new version of the company's code generator product. A Class Model view has been added to clearly display the static model as captured from the modeling tool before and as enhanced after code generation.

The log also contains significantly more trace information,



making template development easier. Other new features include an AND/OR format that renders the filters and conditions simpler and more powerful to use, additional context variables for more precise refinement of templates, and a simplification in initialization and transfer of

metadata from model to code. ☛

www.codagen.com

eHelp Corporation Introduces DynaHelp

(San Diego, CA) – eHelp Corporation, formerly Blue Sky Software, has announced DynaHelp, a new proactive customer assistance solution that significantly improves Web site usability, customer satisfaction and online sales. ☛



www.ehelp.com

IBM Delivers World's First Fully Integrated E-Business Database

(New York, NY) – DB2 Universal Database, Version 7, announced by IBM at PC EXPO 2000, is now generally available. The only database to fully integrate e-business, business intelligence



and content management capabilities, it enables companies to leverage the full value of their data – in many cases at a third the total cost of ownership of other databases. ☛

www.ibm.com/software/data/launch.

Cerebellum Introduces Internet Data Integration Solution

(San Francisco, CA) – Cerebellum Software, Inc., showcased a beta release of version 2.0 at JavaOne. This latest version enables Web-based applications and components to provide real-time access and updates to corporate data distributed throughout the enterprise, and greatly extends the data integration capabilities of leading application servers. ☛



www.cerebellumsoft.com

Protoview

www.protoview.com

KL Group

www.klgroup.com