FOR DEVELOPERS: STEP-BY-STEP PORTLET TUTORIAL

WebSphere® DEVELOPER'S JOURNAL

The World's Leading Independent WebSphere Developer Resource

WebSphereDevelopersJournal.com

MARCH VOLUME 2 ISSUE 3

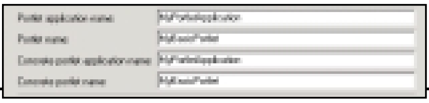
DISPLAY UNTIL MAY 31, 2003
$8.99US $9.99CAN
03>
0 09281 03422 3

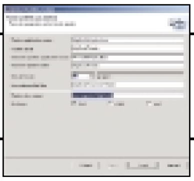
SYS-CON MEDIA

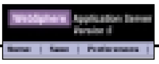
The Evolving Role of WebSphere

The WebSphere marketing team

INTERVIEWED BY JACK MARTIN PAGE 44







Security in Our Virtual World

BY JACK MARTIN

Nothing in life is completely secure. Think about it – when Julius Caesar took that famous walk to work on a fateful March morning, I'm sure he felt rather secure; after all, he was Caesar. Did Caesar think his life was about to end and become the prototype for countless Mafia hits in Hollywood movies and the real world? Of course not. The Internet and all of its technologies are no exception. Thousands of specific security problems have been discovered and fixed since the inception of the Internet. If you're using up-to-date firewalls, application servers, operating systems, and Web browsers – with all of the latest updates and security patches applied as each vendor prescribes – you are usually safe from known attacks. However, nobody is safe from attacks directed at weaknesses that haven't yet been discovered.

If somebody tells you that anything involving technology is 100% safe because hackers aren't smart enough to exploit vulnerabilities, you're either speaking with a liar or a fool. I've discovered some security problems on my own, and I'm pretty sure that I'm not the smartest person in the world. When one group of hackers creates an attack and shares it with other idiots, we're all in trouble.

There are two classes of security problems: nuisances and security breaches. A nuisance attack merely prevents you from getting your work done; for example, it may cause your computer to crash, delete your files, or give you and all of your associates a virus. In most cases nuisance attacks are initiated by the socially maladjusted, the same people you tend to avoid in the real world. These people are angry over being isolated and they blame society as a whole for their personal disappointments and failures. Learning to write computer code that inconveniences people with real lives really appeals to this group.

Security breaches are more serious. There are two types of security breaches. In the first, your private data is stolen and used by the thieves for some type of financial benefit to themselves. The second type is caused by some type of organized group or government that wants to disrupt the economy. The first group – the world's computer-literate criminal class – puts a tremendous amount of time into trying to steal credit card numbers and other data that can be sold on the open market. These are small groups and individuals with limited resources looking for big payoffs. They are primarily the problem of banks, credits card processors, and large corporate data repositories. The thieves who recently breached a security system to get access to about 5.6 million Visa and MasterCard accounts are of this type.

Visa and MasterCard said the thieves could access as many as 2.2 million accounts after breaching the security system of a company that processes credit card transactions on behalf of merchants. The affected accounts make up almost 1 percent of the 574 million Visa and MasterCard accounts in the United States – now that could be a big payoff.

The other type of breach – the kind aimed at disrupting our economy – is everyone's problem. The groups behind such attacks have potentially unlimited budgets to do whatever damage they may want to inflict. We are standing on the verge of the world's first virtual war – a war that will have real-world consequences. Currently there is no easy way to protect yourself from this last group, as they are playing by the real rules of war.

The one thing you should never do is to glorify these groups by calling them hackers, as they wear the name like a badge of honor. Call them what they really are – idiots, fools, thieves, and criminals. As long as there is some perceived benefit to people who cause computer trouble – be it for fun, profit, or political goals – security will remain an issue.

To quote Don Corleone, "This is the business that we have chosen to be in."

ABOUT THE AUTHOR... Jack Martin, editor-in-chief of ***WebSphere Developer's Journal***, is cofounder and CEO of Simplex Knowledge Company, an Internet software boutique specializing in WebSphere development. Simplex developed the first remote video transmission system designed specifically for childcare centers, which received worldwide media attention, and the world's first diagnostic-quality ultrasound broadcast system. E-MAIL... jack@sys-con.com

WebSphere® DEVELOPER'S JOURNAL

ADVISORY BOARD
Richard Arone, James Martin, Greg Williams

EDITOR-IN-CHIEF	JACK MARTIN
EDITORIAL DIRECTOR	JEREMY GEELAN
EXECUTIVE EDITOR	GAIL SCHULTZ
CONTRIBUTING EDITOR	PATTI MARTIN
PRODUCT REVIEW EDITOR	JAY JOHNSON
SPECIAL PROJECTS EDITOR	RICHARD ARONE
MANAGING EDITOR	JEAN CASSIDY
EDITOR	NANCY VALENTINE
ASSOCIATE EDITORS	JAMIE MATUSOW JENNIFER STILLEY

WRITERS IN THIS ISSUE
James Carey, Brent Carlson, Denise Gabardo, Harvey Gunther, Wilfred C. Jamison, Jay Johnson, Jack Martin, Peiyin Pai, Bhavesh Patel, Atit Patel, Hollis Tibbetts, André Tost, John Worrall

SUBSCRIPTIONS
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $8.99/Issue Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)
Back issues: $15 U.S. $20 All others

PRESIDENT AND CEO	FUAT A. KIRCAALI
COO/CFO	MARK HARABEDIAN
SENIOR VP, SALES & MARKETING	CARMEN GONZALEZ
VP, INFORMATION SYSTEMS	ROBERT DIAMOND
VP, SALES & MARKETING	MILES SILVERMAN
PRESIDENT, SYS-CON EVENTS	GRISHA DAVIDA
PRODUCTION CONSULTANT	JIM MORGAN
FINANCIAL ANALYST	JOAN LAROSE
ACCOUNTS RECEIVABLE	KERRI VON ACHEN
ACCOUNTS PAYABLE	BETTY WHITE
ADVERTISING ACCOUNT MANAGERS	MEGAN RING-MUSSA ROBYN FORMA
ASSOCIATE SALES MANAGERS	CARRIE GEBERT ALISA CATALANO KRISTIN KUHNLE
CONFERENCE MANAGER	MICHAEL LYNCH
ART DIRECTOR	ALEX BOTERO
ASSOCIATE ART DIRECTORS	LOUIS F. CUFFARI RICHARD SILVERBERG TAMI BEATTY
WEB DESIGNERS	STEPHEN KILMURRAY CHRISTOPHER CROCE
CONTENT EDITOR	LIN GOETZ
CUSTOMER SERVICE REPRESENTATIVE	MARGIE DOWNS
CUSTOMER RELATIONS MANAGER	RACHEL MCGOURAN

EDITORIAL OFFICES
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9637
SUBSCRIBE@SYS-CON.COM
WebSphere® Developer's Journal (ISSN# 1535-6914)
is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Developer's Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

Don't just consider performance, require it

Practicing Software Performance Engineering

BY WILFRED C. JAMISON

Information technology is changing very rapidly right before our eyes, and so are user requirements. Along with the paradigm shift that resulted from the advent of a worldwide communication infrastructure and the advancement of distributed computing, software performance has evolved from being merely a consideration into a primary requirement. This is especially true for those developing mission-critical applications on the IBM WebSphere Application Server. This article poses a challenge to existing software engineering methodologies to evaluate and check whether they are capable of delivering the expected performance level of the applications under development.

Make It Run, Then Make It Run Fast

As software engineers, we were indoctrinated with this "golden rule." There is indeed a valid reason to adhere to the rule since it emphasizes one basic principle – that every software engineer makes sure that a product fulfills its functional requirements and specifications first. However, there is also an undeniable truth that more often than not, we only satisfy the first part of this rule and rarely the second, which is to make it run fast.

Managing the development of a large software application is never an easy task. That is why the discipline of software engineering was introduced – to offer us a systematic and methodical approach to writing software that is functional, reliable, robust, maintainable, and – most of all – useful to the end users. If you are in the business of writing software that is expected to deliver high performance such as middleware programs (application servers, operating systems, communication tools) or even mission-critical Web applications (financial applications, video conferencing, intelligent search engines, etc.), then this rule is just not going to work. In addition, waiting until an application is running before addressing performance is an expensive proposition.

So implementing the second part of the rule is the harder problem. We software engineers know very well that oftentimes we scramble to meet tight schedules, spending a lot of time understanding somebody else's code, fixing bugs, overcoming unexpected roadblocks, and the list goes on. In this competitive business, there is no time left to make it run fast by the time we get the product up and running.

Make It Run Fast

I propose that we change the rule by making it run fast *from the outset*. In other words, why not do it now instead of later? I believe that this can be achieved only if we treat performance as a function itself, rather than as a feature. The tenet of this article is that we should be able to adjust our existing software engineering methodologies to be more performance-oriented, so that the first running version of the application we write actually does run fast.

Be Proactive, Not Reactive

My discussion is not complete without a description of how most organizations manage performance in their software development process. In simplest terms, I describe this approach as being reactive. Figure 1 is a typical model used in many organizations.

In the reactive model, the traditional software engineering methodology is followed, but any performance-related issues and concerns are discovered only toward the end of the cycle. This is based on the belief that nothing can be measured until a running product is built. Thus, performance work starts typically during system quality assurance testing. At this time, some benchmarking activities are performed by a designated "performance team" test workload. Based on the results, the performance of the application is analyzed and verified against a set of performance requirements or criteria. More often, even after the application is in production, bottlenecks

ABOUT THE AUTHOR
Wilfred C. Jamison, PhD, is a performance analyst and software engineer for the WebSphere Application Server Performance team. He is an expert on memory management and Java performance. He is currently leading the effort on WebSphere Performance for the Linux platform. His other interests include security, Web services, and language processing.

E-MAIL
wjamison@us.ibm.com

are identified and a solution is proposed or written up. There are two possibilities:

- The owner of the component that is identified as the cause of the problem is notified. A solution is proposed or the owner finds a way to fix the problem. The fix is then put into production and the cycle repeats.
- A solution is formulated by the performance team and prototyped to measure its performance gain. Once an acceptable solution is found, it is handed down to the owner for final implementation. Note that during this time, communication between the component owner and the performance team is very important.

Depending on the type and nature of the application, performance analysis can also produce some tuning parameter settings that are used for performance improvements. Tuning is a fundamental practice in performance engineering, especially in a complex system where there are too many interacting components, making it harder to understand or grasp an absolute solution that works for all cases. Also, it is typically appropriate for systems in which the overall behavior depends on the workload coming onto the pipe. Finally, the performance team oftentimes produces a performance report that describes the resulting performance level of the application. It can be used both for end-user reference and as an input to the next development cycle.

This mode of operation is very reactive in nature because technically the performance team's work is to monitor the latest build of the application, raising a flag whenever something is not right and making sure that the problem is fixed. It is reactive because their actions are always in response to what they are given, and there is minimal, if any, influence in the overall design of the application from a performance perspective. Furthermore, the responsibility for making sure that performance is looked after is left to a single person or group.

There are serious problems with this model, in which performance analysis and improvements are done toward the end of the development cycle.

- ***Cost:*** Imagine that a team of sports car engineers design their car to make sure that it actually runs before attempting to make it go fast. Once convinced that it will run, they enhance it with all the necessary high-performance features. Undoubtedly, this is the wrong way of doing it – not just because the car is supposed to run fast to begin with, but because it is a very costly approach. It will cost them time and money, as they have to repeat the process.

Similarly, software often goes through many revisions because of the lack of performance-enhancing features. Consequently, this entails more testing cycles as well as the risk of functional regression. In the end, the product is shipped with poor performance due to lack of time. Revenues go down because of customer dissatisfaction and therefore more hours are spent on improving performance to pump up revenues.

- ***Suboptimal solution:*** Imagine again that our car engineers realized that the performance enhancement they designed cannot be easily incorporated into the car because of the way it was originally designed. (Obviously, they did not think ahead.) So what options do they have? (1) Disassemble the car and redesign it so that the new feature can be accommodated. (2) Forget about the cool enhancement and think of some other performance enhancements.

Similarly, when software is not designed with performance in mind, there is a very high chance that a given performance solution will not be viable because of some architectural restrictions or implementation-dependent constraints – and the cost to revise all these things will be significant in terms of time and money. Thus, the solution is rejected and a weaker alternative is devised. The lesson is that once an architectural design or infrastructure is in place for a sufficiently long period of time, major changes are rarely made. More often we get into the "patches" syndrome where ad hoc solutions are provided until the whole application becomes a crazy quilt of patches.

- ***The tuning trap:*** Tuning has a negative aspect. We have already mentioned what tuning is good for. However, when performance is on the edge of being acceptable and time is ticking away, the last recourse is often to hope that the system can be tuned rather than making application design or code changes. In general, tuning helps, but it does not solve the real problem of the underlying application.

Thus, the better way to approach these problems is to practice software performance engineering and be proactive in providing performance solutions from top to bottom. The software engineer needs to embrace performance as part of the overall requirements and therefore address it right from the very beginning. The car engineers should have made performance the focal point of their design. Like all aspects of the application being developed, project management should be able to track, control, and verify its performance status.

Who Should Address Performance?

Everyone needs to consider performance – analysts, architects, developers, testers, document writers, etc. Thus, everyone is a software performance engineer. Software performance engineering is a concerted effort. It is a methodology that everyone in the organization should get involved with. Figure 2 shows a proactive model of this methodology. As we can see, the original methodology is kept intact with additional bubbles surrounding it.

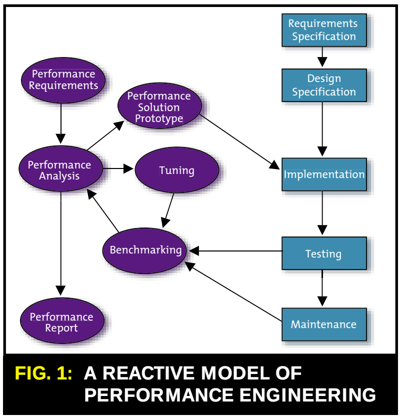


FIG. 1: A REACTIVE MODEL OF PERFORMANCE ENGINEERING

Performance requirements should be identified, to begin with, typically by the architects or the marketing organization. For Web applications, requirements focus on user load and response time, such as how much load the system needs to handle and still respond in no less than 5 seconds. The key steps are:

1. Define the performance requirements of the application, in both qualitative and quantitative terms.
2. Define a set of measurable criteria to verify these performance requirements.
3. Define a set of specific test scenarios to use during the test stage.

Once the performance requirements are put in place, designing the architecture of the application and the individual components must involve viewing them from a performance perspective. Typically, an experienced engineer with expertise on performance should be included in the design process. Another very important inclusion in this model during the design and implementation stages is the consideration of current performance technologies, such as caching, performance design patterns, and fast algorithms. An expert in this field must be identified to help in the process.

I consider performance analysis to be the most important part of this model. It is done throughout the development cycle. During the design review, for example, a design document specification must be given to a group of performance analysts for the purpose of catching potential bottlenecks. They may also provide recommendations on how to modify the design for better performance.

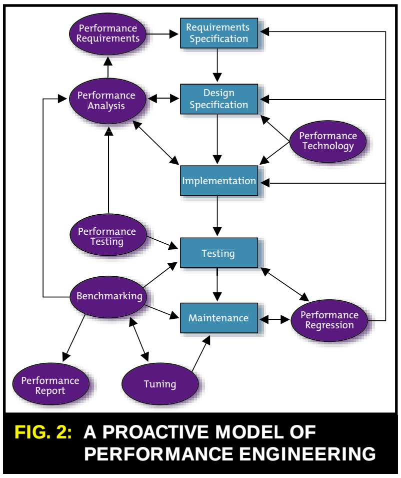


FIG. 2: A PROACTIVE MODEL OF PERFORMANCE ENGINEERING

Performance experts typically look for problems like communication pathway and protocols, path lengths (both for design and implementation), resource requirements such as memory and CPU cycles, critical nodes, etc. If the operational environment is also specified, they also analyze the characteristics of the components in the environment such as the network bandwidth, storage devices, gateways, etc. Thus, the expertise is very crucial in performance analysis. The performance analysts may also go back to the requirements specification if necessary.

During the implementation stage, developers should be made aware of the significance of high-performance programming. They need to be trained to learn and think of fast algorithms and to use any language-specific features that can boost performance. Also, it will be productive to provide them with a list of best practices for high-performance application programming, especially in the areas of Java and J2EE. The performance code review also includes performance analysis (not necessarily by the same group that did the design review). This group is composed of engineers with excellent technical skills who can pinpoint poor algorithms as well as recommend better solutions. This process should be conducted repeatedly until all are satisfied.

Testing is an integral part of the performance engineering process. Performance-specific test scenarios must have been defined early on and carried out during this stage. One important thing that should be mentioned is that every functional unit must be tested for performance. Thus, testers and developers are trained to do performance testing and to use tools such as profilers and performance analyzers. They should also be trained to perform microbenchmarking on their own code. This leads me to my principle that performance is distributable; that is, it can be modularized such that performance of every unit contributes to the whole.

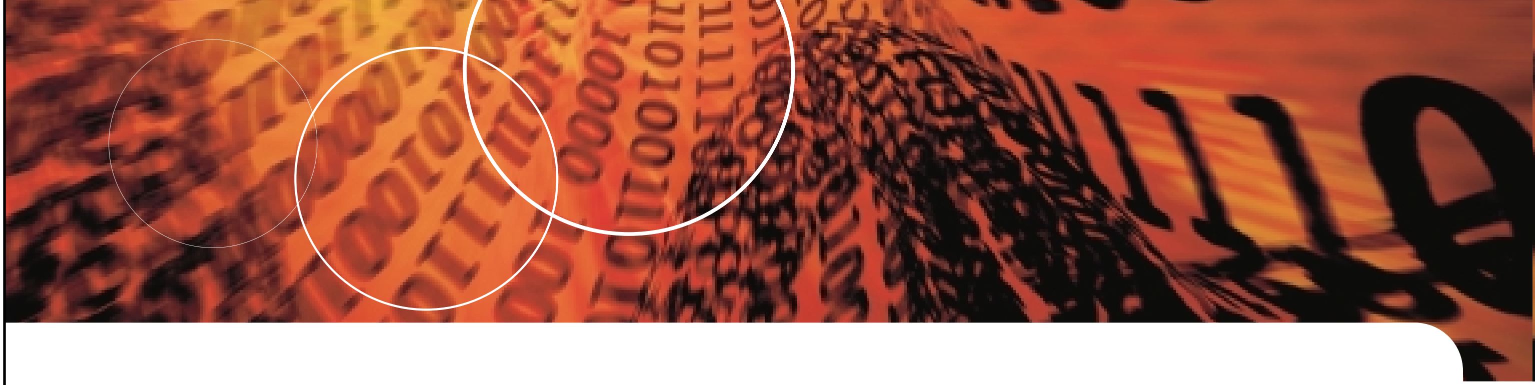
Benchmarking can be done once a running product is built. In both benchmarking and system testing, performance analysts may get involved once again to help diagnose problems found. Any problems and/or solutions should be communicated either at the design or implementation level, depending on the case. Performance problems encountered should be documented and monitored by management.

Benchmarking is a discipline in itself and there are numerous things that can be written about it, such as the proper way of conducting benchmarks. I will not discuss them in this article. Suffice it to say that benchmarking efforts do help in discovering the problems and weak points of the application. Also, benchmarking helps to keep track of where the application is at in terms of performance at any given point in time. Typically, benchmarking results in performance reports as well as tuning guidelines.

Another important aspect of testing, as well as maintenance, is performance regression. As the application evolves and different builds are created, performance must be monitored to see if any performance regression occurs. Regressions need to be reported to the appropriate people – designers, analysts, developers, architect, etc. – and corrected.

The Bottom Line Is Team Performance

Contrary to what others may think, software performance is a broad area – it encompasses issues such as scalability, high availability, and capacity planning, to name a few. Although the metrics may be as simple as throughput and response time, there are hundreds of factors that affect performance. The responsibility of making sure that good performance is achieved belongs to the entire organization. By executing performance analysis early on – from requirements analysis to maintenance – we cut down the total development cost by catching potential problems early. Another advantage is that we are able to design the architecture in a performance-oriented manner and therefore we get the optimal performance boost. Making changes later on will not be as hard.



Application Assembly Platforms

Untangling the web of enterprise application development

– BY HOLLIS TIBBETTS –

For enterprises to achieve the full potential of WebSphere, higher-level software architecture needs to be combined with best practices design methodology. An argument can be made that the application assembly platform will become the next stage in the evolution of enterprise application development.

Application assembly platforms will be the next generation approach for building enterprise Web applications based on a service-oriented architecture. An application assembly platform provides an integrated suite of technologies that allows organizations to visually assemble new applications that leverage enterprise data resources, legacy systems, packaged applications, J2EE components, and Web applications. Application assembly platforms are designed to leverage the full power of J2EE, while masking its complexity – effectively turning Java developers into J2EE experts.

What are the driving factors behind the need for an application assembly platform? It's simple. Enterprise Web applications are typically over budget, expensive to maintain, and offer minimal reusability. In fact, most J2EE shops today report the following problems:

- Limited code reuse
- Lack of productivity during development and ongoing maintenance phases
- Difficulty in integrating with back-end systems and in leveraging existing J2EE/Web investments
- Business users unable to effectively participate in development
- J2EE experts required at all levels of development, exacerbating skill shortages
- Performance/scalability problems

The truth is that although J2EE application servers such as IBM's WebSphere have enabled an entire generation of sophisticated Web applications to be built, most organizations fail to tap into WebSphere's full capabilities. A new approach is needed to dramatically lower the bar for creating enterprise-quality applications.

ABOUT THE AUTHOR
Hollis Tibbetts is VP of product marketing for M7 Corporation. He has over 17 years of experience in a variety of technical and marketing roles at companies including Sybase, OnDisplay, and webMethods.

E-MAIL
hollis@m7.com

Web Application Development Today

The barriers to fully realizing J2EE's potential fall into two basic categories: those resulting from the inherent complexity of the J2EE specification and those relating to internal organizational issues.

J2EE COMPLEXITY

Despite, or perhaps because of, its rich functionality, J2EE is an extremely complex system with a steep learning curve. Furthermore, mastering J2EE really means mastering complex underlying Java standards such as JSP, EJB, JDBC, and JMS. Web services has added to that complexity – requiring programmers to understand XML, SOAP, and WSDL in addition to Java. All these pieces require an additional level of skill to integrate effectively.

As Java has evolved from a programming language into the J2EE platform, it has expanded in richness and functionality. Indeed, the J2EE Specification has grown from 250 pages in J2EE 1.2 to 1,753 pages in J2EE 1.4. This does not include JMS (140 pages), JCA (290 pages), or anything about Web services, which are not part of the specification. In addition, when building enterprise Web applications, developers are confronted with a variety of approaches, such as JavaBeans, EJBs, and Web services, as well as a variety of heterogeneous data sources that require sophisticated integration efforts.

So while a J2EE application server provides many built-in capabilities, most organizations find that it is taking them far longer than they expected to master these capabilities.

INTERNAL ORGANIZATION ISSUES

Other barriers to full realization of the benefits of WebSphere manifest themselves within the walls of an organization. Some common ones are:

- ***Skill profiles:*** Most enterprises have many more business-level developers than skilled J2EE experts. According to Gartner, 71% of survey respondents named a lack of qualified resources and skills as the top problems in Java adoption.
- ***Visibility:*** Traditional J2EE development approaches are

completely opaque to average business users, yet may have key knowledge crucial to the development and maintenance of critical business processes.

- ***Leverage of existing investments:*** Few organizations are able to create application architectures that fully leverage their existing investments in data, Web applications, Java components, and other existing systems.
- ***Reusability:*** Reusability of new development is a key component of achieving the rapid deployment of applications, and is central to reducing long-term costs, yet few organizations are able to achieve it.
- ***Integration:*** Almost all modern applications need to integrate with multiple data sources, applications, Web services, and existing J2EE components. Until now this has often been a painful and expensive process.
- ***Maintenance:*** On average, companies spend at least 30% of their software budget on maintenance and support. More than 1 company in 10 spends over half of their software budget on maintenance, according to a 2002 survey by Evans Data Corporation.

A recent Gartner report...states that by 2006 adoption of products such as application assembly platforms "will increase by a factor of 10, due to increased awareness of and usage of service-oriented architecture and designing for application integration"

These barriers conspire to drive up the cost and time to develop enterprise-class Web applications.

HOW WEB APPLICATIONS ARE DEVELOPED TODAY

To fully appreciate the potential of application assembly platforms, it is helpful to look at how enterprise Web applications are developed today. To build a robust and scalable Web application, the logical choice is to base it on a J2EE platform such as WebSphere.

Most enterprise-class applications access a variety of existing back-end systems. These systems run the gamut: databases, EAI servers, custom frameworks, messaging systems, legacy applications, and packaged applications. To expose these back-end systems through WebSphere, organizations need to develop components – EJBs, entity beans, session beans, Web services, JavaBeans, etc. These "existing assets" are required to run a Web application, but they obviously don't make a Web application by themselves – that requires a presentation layer, which is typically HTML or more commonly JSP (JavaServer Pages).

The next step is to access the existing assets from a Web page, but it's more complicated than just dragging and dropping an object onto a Web page. If, for example, you want to interact with an existing asset from a JSP that you're developing, you must know the details of the component in order to access it. You need to be fairly knowledgeable about the nature of the component you're calling. For example, if you're calling a Web service, you need to create a SOAP envelope to call the service, marshal the return information, and so on. Furthermore, you've hard-coded this logic into the JSP page, making it harder to maintain.

To further complicate the situation, Web applications contain from dozens to hundreds of different pages. The logical and conditional flows between these pages represent business processes. The links between all these pages are typically hard-coded, with a significant amount of conditional logic embedded into the Web page. Each of those individual pages contains custom code in the form of Java, JavaScript, or other languages. And each page in turn has hard-coded links into existing back-end assets. The result is a lot of "spaghetti code," a problem that becomes compounded as the spaghetti code is distributed and duplicated across a large number of pages (see Figure 1).

Finally, for a Web application to solve a typical business problem, 50–100 business processes need to be designed and followed. Ironically, the developers working on the application are often the least informed about such processes; instead, the processes are owned by the business analyst or domain experts. However, the complexity of J2EE solutions has rendered business analysts unable to contribute their knowledge in the most effective manner.

PARTIAL SOLUTIONS

In the few past years, several partial solutions to these problems have been offered. Most enterprises rely upon Java IDEs and Web-page editing tools to develop complex Web applications. These tools provide increased developer productivity, but because of limited scope fail to address some of the larger architectural problems.

Traditional approaches to productivity are usually in the form of code libraries, code generators, and RAD (rapid application development) tools. While these can be helpful in some cases, they are not without problems. Developers have been disappointed with CASE (computer-aided software engineering) and RAD tools because of their inflexibility in accommodating round-trip hand-code editing and their lack of support for the special application requirements of each enterprise. Typical CASE modeling tools and IDEs fail to fully address the complexity of J2EE; they automate low-level details, but do little to help provide a proper application architecture or methodology optimized for J2EE.

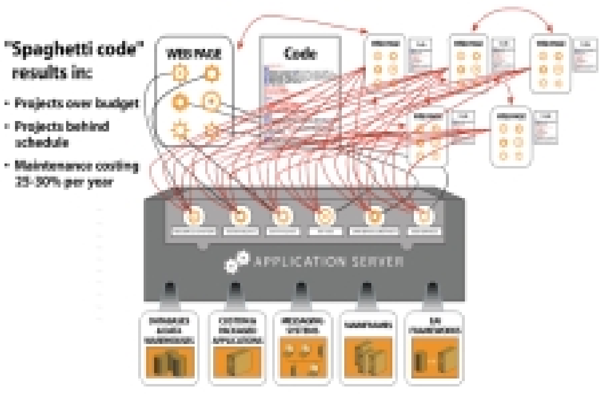

"Spaghetti code" results in:
• Projects over budget
• Projects behind schedule
• Maintenance costing 25-50% per year
WEB PAGE
Code
APPLICATION SERVER

FIG. 1: SPAGHETTI CODE

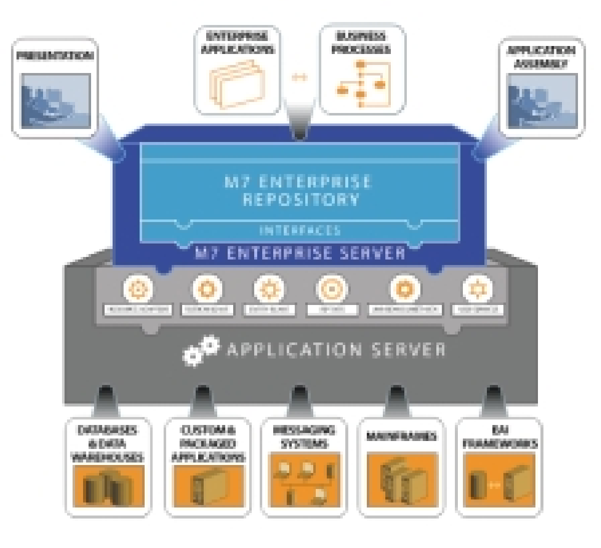


FIG. 2: THE APPLICATION ASSEMBLY PLATFORM

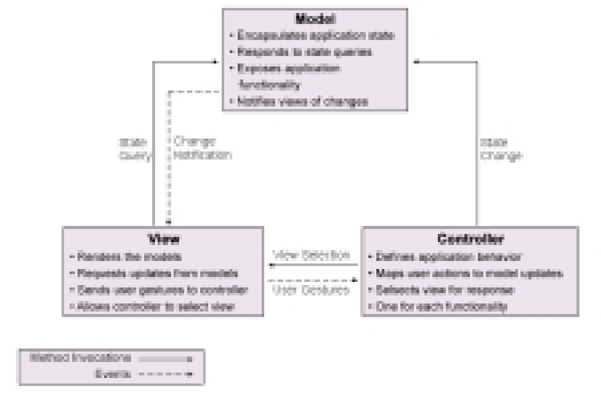

Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

FIG. 3: THE M7 APPLICATION ASSEMBLY PLATFORM USES AN EXPLICIT VISUAL MODEL TO REPRESENT THE APPLICATION WORKFLOW AND BUSINESS LOGIC

Many organizations have tried to build their own in-house libraries or code generators, only to find that they are difficult for all but a few to learn. In addition, there are ongoing costs associated with maintaining the libraries and tools.

Struts is an open-source framework for implementing large-scale applications. It includes a set of custom tag libraries and their associated Java classes, along with various utility classes. Still, Struts is appropriate only for highly skilled J2EE experts and fails to address the broader issues of application maintenance.

The Application Assembly Platform

With the broad adoption of WebSphere and other J2EE application servers, there is now a need for products that can reduce the complexity of these systems and boost productivity for companies building, maintaining, and deploying these applications. This is validated by a recent Gartner report, which states that by 2006 adoption of products such as application assembly platforms "will increase by a factor of 10, due to increased awareness of and usage of service-oriented architecture and designing for application integration."

APPLICATION ASSEMBLY PLATFORM FUNCTIONALITY

Broadly speaking, an application assembly platform's features have the following goals:

- To promote superior application quality
- To reduce development costs by increasing developer productivity
- To reduce maintenance effort and costs

An application assembly platform accomplishes this by providing:

- A component-based architecture, design methodology, and technology that can be used by all levels of IT developers, from Java experts to business analysts
- Automation for many of the common and time-consuming aspects of Web application development
- A platform where best practices and quality are inherent to the systems built

KEY CONCEPTS

A number of concepts behind the application assembly platform help it achieve "more with less." These concepts are:

- ***Abstraction:*** Abstract the developer at the visual and business-process level from the underlying complexities and implementation details of the objects being accessed. As a result, the developer at the visual or business-process level is completely unaware of how a particular piece of data is implemented (e.g., an entity bean, a session bean, an EJB, or a Web service. Access to these objects is not simply made easier, it is made completely transparent. By doing this, the application assembly platform has provided an "automatic" best practice: data access code has been completely removed from the presentation layer.

- ***Repository:*** Allow objects to be defined, shared, and reused. The repository is the key enabling technology that allows the application assembly platform to abstract developers at the visual and business levels from the details of object access. The repository not only stores and describes J2EE objects, but also business process definitions and workflows. Ideally, this repository should be implemented in an open format, such as XML.
- ***Interfaces:*** Allow access to existing and future components without requiring them to be changed or "wrappered." Key to the concept of the interface layer is the idea of "if you can describe it, you can access it." For optimal performance, the interface layer is thin, essentially a set of descriptors that allow direct access to the underlying J2EE components.
- ***Business process automation:*** Allow business processes to be visually represented at development and automated in the J2EE server at execution time. This is crucial to maximizing productivity and reuse. Visually representing the business processes and the conditional flow of the Web application allows the developer to remove the "spaghetti" code from the Web pages themselves. This accomplishes the goal of layer separation: the Web pages themselves contain only presentation. The business logic (along with the data access logic, courtesy of the abstraction layer) has been removed from the Web page, in keeping with accepted best practices. At execution time, these business processes are run in WebSphere and are implemented as standard JavaBeans.
- ***Visual assembly:*** Provides a visual paradigm that allows for the visual assembly of Web pages and Web applications. This final capability allows the creation of the presentation layer of the application (i.e., the HTML and JSP pages). By having a visual environment that is cognizant of the contents of the repository as well as of the various business processes in play (to provide context and scope), visual assembly allows for the rapid creation of Web pages that have scope and context. This is typically enabled by the automatic creation of tags that are populated at execution time by WebSphere. The result of this is Web page and Web form creation at many times the speed of other methods.

KEY BENEFITS

The application assembly platform (see Figure 2) delivers on its promise of compressing application development time and investment.

Separation of Data Access, Business Logic and Presentation

Although the separation of business logic from the presentation layer has long been recommended, in Web applications it has been very difficult to achieve. The code at each layer is so low-level that there is no way to represent higher-level business processes. All too often, business logic, presentation, and even data access are implemented with a tightly intertwined mixture of Java, JavaScript, HTML, and other language fragments.

The application assembly platform has an explicit mechanism that makes it easy to enable the clean separation of layers and the reuse of business processes, business rules, and objects; the AAT offers an abstraction layer and an object repository and business process automation layer. This makes it much easier to build and maintain applications and ensure consistency.

Automation of Application Workflow and Navigation

One of the areas that is most problematic in maintaining Web applications is routine changes to accommodate the application workflow and navigation, which can comprise business processes. Unfortunately, the only representation of the application workflow is usually found on a whiteboard or in some cases, a UML diagram. When the application is implemented, the workflow and navigation logic is hard-coded into a combination of HTML links and JavaScript.

To solve this problem, my company, M7 Corporation, offers the M7 Application Assembly Platform, which uses an explicit visual model to represent the application workflow and business logic (see Figure 3). By working visually, changes to workflow are handled in minutes without any need for low-level coding. At execution time, the visual representation is executed within the WebSphere Application Server as standard JavaBeans. Similarly, the AltoWeb Application Platform uses process flow diagrams and prebuilt components to visually define business logic. The Agentis Software's AdaptivEnterprise Suite takes a slightly different approach, focusing on the creation of highly adaptable applications. By combining business-level service model design, an application transformation system, and an agent server that can retry different processes at runtime and dynamically flag situations where no discrete outcome is available, Agentis delivers applications designed for highly dynamic environments where unexpected business events must be managed.

"An understanding of J2EE, JDBC, SQL, and transactions is no longer needed to implement a simple dynamic"

Reuse of Business Processes, Rules, and Objects

Although reuse has long been a goal of the software development industry, it has really only taken hold at the most basic level with code libraries. The M7 application assembly platform includes a repository (see Figure 4) that enables the reuse of business processes, rules, and objects across applications.

The goal is that everything becomes a reusable component: components, workflows, business processes. Even presentation pages (which have been separated from their business logic) can have input and output parameters, allowing for a functional style of programming in the presentation layer. This means that a screen designed for one application can be easily reused in another application. Bowstreet Factory takes a slightly different approach, in which developers rapidly build Web applications by pulling together a sequence of highly adaptive, reusable software components

“An application assembly platform will use EJBs in a highly optimized fashion to ensure that server processing remains efficient regarding both CPU and memory requirements”

called Builders. Developers assemble Builders into models, similar to the way they would build a spreadsheet model by snapping together formulas. These models are then executed at runtime to dynamically generate application code, including JSPs, Java classes, and XML documents, as well as all of the low-level artifacts that make up the application.

Organizations frequently have substantial bodies of pre-existing code that works, and works well. The application assembly platform is designed to integrate with pre-existing code, rather than forcing substantial portions of legacy functionality to be rewritten. Ideally, the application assembly platform “exposes” existing objects, and does not require wrappers.

Consistent Access to Enterprise Data

One of the challenges in most large businesses is that they have a wealth of data stored in a broad range of systems. Each of these has its own unique way of representing and accessing data. For application developers, this represents a considerable learning curve and an obstacle to tapping into existing resources.

The application assembly platform’s abstraction, repository, and interface capabilities make it easy to incorporate existing objects – whether JavaBeans, relational data, legacy systems, or Web services – into your applications. As a result, objects can be used independently of their underlying physical representation. This abstraction of objects allows almost anything – an EJB, a Web service, etc. – to be invoked by the developer without having to worry about the underlying protocol and implementation details.

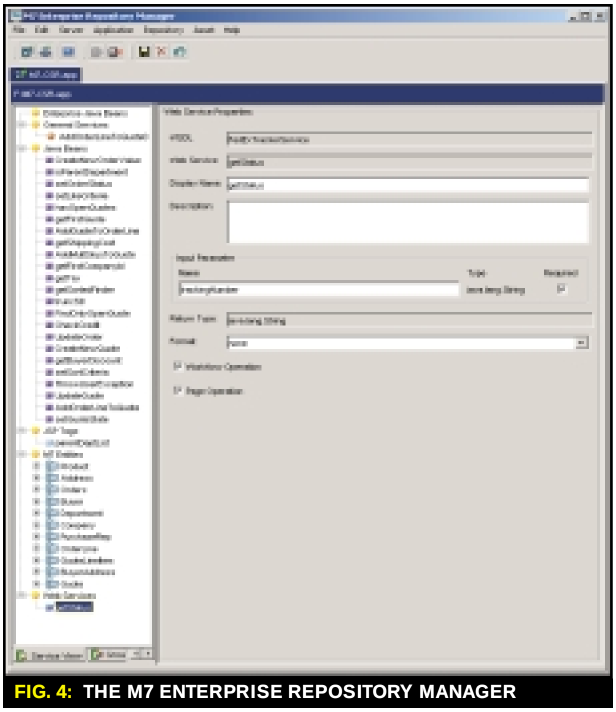


FIG. 4: THE M7 ENTERPRISE REPOSITORY MANAGER

This approach provides a much-needed layer of insulation that greatly simplifies code reuse and maintenance. As the data access layer encapsulates the underlying access, the data can be used in applications independently of its physical representation. If the underlying physical representation changes, the higher levels of the application are insulated from the changes and require no additional changes.

Optimization of J2EE Resources for Performance and Scalability

Many early users of J2EE have struggled with optimizing performance, especially when using EJBs. An application assembly platform will use EJBs in a highly optimized fashion to ensure that server processing remains efficient regarding both CPU and memory requirements. In addition, an application assembly platform can implement caching so applications automatically benefit from faster reads and writes, eliminating a common bottleneck in applications. This is a good example of the type of best practices that can easily be “baked into” applications created with an application assembly platform.

Allowing All Types of Developers to Be Productive

The abstraction provided by the application assembly platform’s repository allows all levels of developers, including application developers and Web designers, to contribute to an application. An understanding of J2EE, JDBC, SQL, and transactions is no longer needed to implement a simple dynamic. It is critical for enterprises to leverage all their resources in building large and complex Web applications for the J2EE platform.

Summary

In 1996, state-of-the-art Web development was a hodgepodge of CGI-bin scripts held together with Perl. The emergence of WebSphere as an application server provided the foundation for a more robust, enterprise-quality platform for delivering applications. Application servers made the truly difficult possible, by incorporating key elements for transaction support, distributed computing, load balancing, and so on.

Most organizations, though, fail to achieve the full potential that WebSphere offers them; productivity, maintainability, reuse, and other metrics of best practices show little progress over previous generations of technology.

The application assembly platform will soon offer an easy way for all developers to tap into the knowledge and expertise previously available only to J2EE gurus so that applications incorporate best-practice design patterns automatically.

Acknowledgment

Tim Lopez, a software architect for M7 Corporation, provided assistance in writing this article.

Part 2: Refining the candidate pattern

Discovering and Documenting Business Application Patterns

BY BRENT CARLSON
AND JAMES CAREY

This month we'll look at pattern discovery in more detail by continuing to examine a business application pattern we discovered and documented. Our focus isn't on creating formal patterns, but on capturing, refining, and sharing the knowledge gained during development. We'll discuss the steps we went through as we discovered and captured our pattern. As we progress through the steps, we'll show you not only what we learned as we captured this pattern, but also things we learned as we captured other patterns.

ABOUT THE AUTHOR

Brent Carlson is vice president of technology and cofounder of LogicLibrary, Inc., a leading provider of enterprise software and services for enabling existing software development assets as Web services. With James Carey, Brent has coauthored *SanFrancisco Design Patterns: Blueprints for Business Software*, and *Framework Process Patterns: Lessons Learned Developing Application Frameworks* (Addison-Wesley).

E-MAIL

brent.carlson@logiclibrary.com

In Part 1 we identified the problem – managing configurable balance information in the area of warehouse management, and the candidate pattern – encapsulating the criteria for the balance information in the ProductBalanceMapKey class, as shown in Figure 1.

Now we'll look at the refinement of the candidate pattern as we applied it to the problem of managing configurable balance information in the area of financial accounts. In Part 3 we'll look at applying this pattern (with others) to the construction of applications, components, and Web services.

The Second Problem

The second time we encountered the problem of managing configurable balance information was in the area of financial accounts. We needed some way to manage cached account balances – the value of a particular account (or set of accounts) for a particular set of criteria. For example, when you go to the bank to check on your savings account you want to be able to quickly get the balance – you don't want to wait while the bank adds up all of the transactions for your savings account. While this example is simple, the requirements for account balances in general are quite complex.

One of the first obstacles we had to overcome was the fact that the domain experts talked about financial account balances in terms of a set of base balances – the balance of each account for each period. They didn't think the balances needed to be configurable. In this case, an implementation had, over time, become their view of the requirement. Once we overcame this problem we discovered that the true requirement defined a need for even more configurability than was the case for product balances. The requirement was that balances for a set of complex criteria needed to be retrieved rapidly. Noncached balances would either be derived from existing balances or would be calculated from the raw transaction information. Understanding the actual requirement enabled us to provide a solution that allows the user to maintain just the specific balances they need for their particular business and to add more as needed. This means that the application (or customer) can make the trade-off between storage and update cost versus time for retrieval.

The Solution

Applying these requirements to the candidate pattern led us to refine it and also to identify another pattern. We're going to focus on what happened to our candidate pattern.

Just as was the case with product balances, account balances have a number of criteria over which balances could be kept. The account balances criteria included:

- ***Account:*** The requirement was that balances be specified at the granularity of the codes that make up the account. For example, an account could be made up of the following codes: major code, minor code, product, and salesperson. Balances could then be specified for a subset of salespeople and a subset of the products.
- ***Prime currency:*** The currency in which the transaction occurred. The requirement was that individual currencies or sets of currencies could be specified.
- ***Fiscal period:*** The year can be broken into multiple fiscal periods. For example, the year may be broken into four periods (one per quarter). The requirement was that balances could be specified for individual periods, sets of periods, and ranges of periods.

For illustrative purposes we'll just look at the account as a whole (i.e., ignore the codes). For example, we could have a set of financial transactions like those shown in Table 1.

Our prior solution is nearly sufficient here – change CachedProductBalances to CachedAccountBalances and ProductBalanceMapKey to AccountBalanceMapKey. The key difference is that for account balances we need to be able to do more than simply turn criteria on and off (i.e., an inUse array on CachedAccountBalances isn't enough). Instead we need to be able to specify the criteria in more detail. For example, we want to be able to cache the balances for a specific account (Savings) or accounts (Savings and Checking), rather than be forced to cache balances for every single account, as would be the case if we reused our original solution for product balances. We may even want to cache them for specific ranges of criteria (e.g., fiscal period 1 through 2).

In Part 1 we introduced the inUse array to determine, when updating and retrieving cached balances, which criteria were applicable and to replace the unused criteria with a placeholder. In this case we need a more complex array that can handle the transformation and management of the criteria. However, we still want our CachedAccountBalances class to do this without becoming entangled in the specific criteria being used. We met this requirement by introducing something called a specification key – a key that specifies a set of criteria. In order to be able to talk about the other keys (such as the AccountBalanceMapKey) we identified these as access keys – keys used to access information. The specification key contained the specification for each criterion and could transform an access key for use with a particular collection of cached balances.

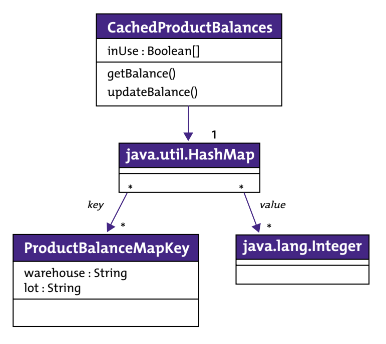


FIG. 1: THE CANDIDATE PATTERN

ACCOUNT	FISCAL PERIOD	CURRENCY	AMOUNT
Savings	1	USD	+100
Savings	3	USD	-50
ForeignSavings	1	SEK	500
ForeignSavings	2	SEK	250
Checking	1	USD	1000

TABLE 1: SAMPLE FINANCIAL TRANSACTIONS

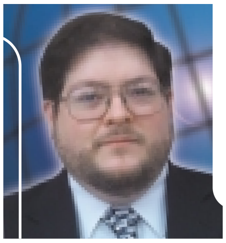
At this point we identified the cached balances candidate pattern. In this pattern, as shown in Figure 2, we have a CachedBalanceSet that contains a set of balance values (CachedBalance) associated with access keys (AccessKey) that are specified by a specification key (SpecificationKey). In order for the CachedBalanceSet to be able to work with the keys without knowledge of the details, AccessKey must support equals and SpecificationKey must support convert (taking an AccessKey and returning a converted one). In other words, AccessKey and SpecificationKey declare interfaces that at this stage in the definition of our candidate pattern would then be implemented by domain-specific subclasses (e.g., a SpecificationKey subclass that supports specifying financial account balances, and an AccessKey subclass that is used to manage the various account balance criteria such as fiscal period and account code).

At this point, our first thought was to tie the key pattern and the cached balance patterns together. However, we ultimately chose to separate them into two distinct patterns since we have found that it is easier to later combine patterns versus trying to split them apart. In fact it may turn out that one of our candidates is in fact a pattern while the other isn't (or isn't worth turning into one). In our case we kept them separate because the key pattern – allowing algorithms to work with criteria abstractly, and the cached balance pattern – maintaining balance information over a specified set of criteria, met clearly different objectives and requirements.

Focusing on the key pattern, we realized that while each key (access and specification) could be written for a particular use of the pattern, we could provide an implementation of the core part of the function. (Such an implementation is often called a mini-framework or framelet.) To do this we created a new abstraction. This new abstraction, called a keyable, allowed the key to work with its criteria generically, thus eliminating the need for domain-specific implementations of AccessKey and SpecificationKey. An AccessKey was made up of AccessKeyables and a SpecificationKey was made up of SpecificationKeyables.

Each keyable defined an abstraction that declared the methods used for AccessKey and SpecificationKey. For example, AccessKey supports equals by comparing the AccessKeyables in the same position by using their equal method. If every AccessKeyable is equal, then the entire AccessKey is equal. Specific AccessKeyable subclasses provided their own implementations of equals to support this algorithm. This change to our candidate pattern had a nice side effect – by transferring the domain-specific responsibility from the Key classes to the Keyable classes, we were able to create a group of generally reusable Keyable classes (e.g., DateKeyable, PeriodKeyable, StringKeyable) that could be assembled to meet many different domain requirements. However, this approach also led us into a trap that we'll discuss below.

In this particular case, we decided to provide a reusable implementation of the pattern. In doing so, we provided a family of keyable subclasses that handled the majority of the different types of criteria. This was both a good and a bad idea. It



ABOUT THE AUTHOR

James Carey, senior software engineer, is lead architect for IBM's WebSphere Business Components project. This project develops EJB components that provide business content to support application development. With Brent Carlson, James has coauthored *SanFrancisco Design Patterns: Blueprints for Business Software*, and *Framework Process Patterns: Lessons Learned Developing Application Frameworks* (Addison-Wesley).

E-MAIL
jecarey@us.ibm.com

was a good idea because providing prebuilt, partial implementations helped to enforce consistency in the use of the pattern; it allowed all uses of the pattern to take advantage of performance improvements in the base implementation; and it made the pattern attractive – a lot of the work was done for you. It was a bad idea because we ended up capturing and documenting the implementation and not the pattern. This was a problem because it made it very difficult to adapt the pattern to new situations – instead of adapting the pattern to the problem, we ended up adapting the problem to the pattern. In our case we exacerbated the problem by naively documenting the implementation as if it were the pattern. Had we clearly separated the pattern and implementation from the beginning, it would have been much better.

Once we identified this pattern, we needed to go back and determine if we should update our implementation of the CachedProduct-Balances. The main thing to keep in mind is that it is not necessary to make this sort of update. Rework takes time, so assume you won't make the update and convince yourself you should. While there are many considerations when making this decision, we found that the key ones were:

- ***Pattern completeness is a key consideration:*** Make sure that the pattern is refined enough that you won't have to go back and reapply it again.
- ***Consistency is a very good reason for going back and making the change:*** Consistency makes code easier for customers to consume (especially in the case of a framework) and easier to maintain. However, in many cases it is better to leave completed code alone and use the pattern from this point forward.
- ***Unrealized requirements occur when the original requirements aren't complete:*** Often we would go back to the domain experts with the refined pattern and explain the refinements added due to the new application. This would make them realize that either they hadn't completely stated the requirements, or they had limited their requirements by what they thought was possible (i.e., what had been done before). Be careful, though, that you don't influence the answer, since you don't want to add unnecessary complexity.
- ***Risk includes a number of things:*** One is the pattern completeness described above. Others are things like your development schedule and development team. It is never worth missing a critical deadline just to replace completed code!

Remember that a pattern may never be complete. It may continue to be refined as each new use is found. Even in our example, where we spent a lot of time refining, documenting, and implementing the pattern as part of the second use, we didn't identify everything. As we continued on the project, we not only found other similar uses, but also found new and unexpected places where the pattern could be used.

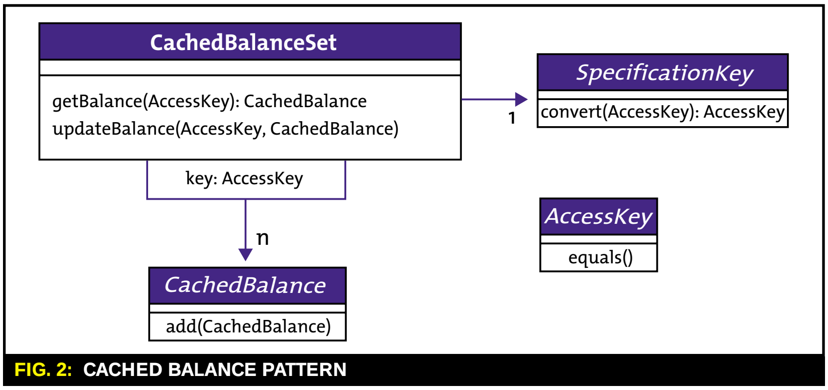

CachedBalanceSet
getBalance(AccessKey): CachedBalance
updateBalance(AccessKey, CachedBalance)
key: AccessKey
n
CachedBalance
add(CachedBalance)
1
SpecificationKey
convert(AccessKey): AccessKey
AccessKey
equals()

FIG. 2: CACHED BALANCE PATTERN

We discovered that in many of these new cases we didn't have to update the core pattern, but instead added implementation notes that addressed applying the pattern in these particular cases. However, keep in mind that the capture of workaday patterns doesn't require formal documentation. Document only as much as you have to. Document enough to ensure you don't lose what you've learned and enough to enable the intended pattern consumer to benefit from what you've learned. You can always formalize it later.

Summary

- Make sure you are working with requirements and not a particular implementation.
- Watch out for new candidate patterns. Don't combine candidate patterns too early.
- Don't forget that your candidate still may not be a pattern (or a pattern worth capturing). You may need to wait for another occurrence to prove or disprove this point.
- Be careful about providing reusable implementations of patterns (i.e., framelets).
  –Don't forget that the implementation is not the pattern.
  –Don't forget you can use the pattern without using the implementation.
  –Clearly document what is the pattern and what is the implementation.

- Ask yourself, before reapplying the pattern
  –Is the pattern ready to be reapplied?
  –Do you need to reapply for consistency?
  –Does the refined pattern expose unrealized requirements?
  –Is it worth the risk in delaying and/or destabilizing existing code?

- Recognize that refinement of the pattern may continue indefinitely.
- Document only as much of the pattern as is necessary.

Now that we've refined and captured the pattern, next time we'll look at applying the key pattern and cached balances pattern (with others) to the construction of applications, components, and Web services.

Optimizing WebSphere 5.0 Performance

Using EJB 2.0 caching and Read-Ahead hints

– BY HARVEY GUNTHER –

ABOUT THE AUTHOR

Harvey Gunther, a senior software engineer in IBM's WebSphere Technology Institute, is currently working on the performance of XML, Web services, and XSL technologies in WebSphere. Harvey has 15 years of server runtime performance analysis and development experience with IBM products.

E-MAIL

hgunther@us.ibm.com

IBM WebSphere Application Server (WAS) 5.0 introduces EJB 2.0 support, which offers new and important opportunities for application architects, such as EJB 2.0 local interfaces; container-managed relationships for EJB-modeled associations; message-driven beans; EJB home methods; EJB-select methods; an internal EJB finder; and EJB-QL, a standard way of defining EJB select and EJB finders. All of these are part of the J2EE EJB 2.0 Specification. To improve the performance of container-managed persistence entity beans WAS 5.0 offers two new tuning levers that reduce database latency and can significantly improve application performance.

- ***Lifetime in Cache:*** Offers improved caching of EJB entity bean data.
- ***Read-Ahead:*** In container-managed relationships (CMR), use of this option allows association data to be retrieved from the database simultaneously with the parent bean.

Caching entity bean data is an optional optimization that provides for the one-time loading of the bean's persistent state. Subsequent transactions use the bean's state without causing a refresh from the underlying data store. Figure 1 illustrates this. In Transaction 1, the EJB Client causes Session Bean1 to retrieve Data Value A from EntityBean1, which is newly loaded. With the caching of entity bean data, Data Value A is loaded from the database and cached during Transaction 1. When the EJB Client starts Transaction 2 to get Data Value again, EntityBean1 and Data Value A are already loaded and cached. There is no need for an additional trip to the database.

We will discuss the performance impact of the Lifetime in Cache and Read-Ahead options through performance primitives and light application functionality performance test cases designed to isolate maximum performance impact. We also examine Lifetime in Cache based on two example applications to put the performance benefit in perspective.

Figure 2 shows the system topology and application architecture for the performance analysis in this article. One key element is that in all cases the database, DB2, is on a separate server.

Lifetime in Cache

This new and optional feature applies to EJB 2.0 container-managed persistence (CMP) entity beans only. Lifetime in Cache caches CMP bean state data and provides for time-based invalidation based on one of the following application-configurable criteria:

- ***Elapsed time:*** Wall clock time set in milliseconds
- ***Clock time:*** Time of day as an offset from midnight
- ***Week time:*** Day of the week as offset from midnight Sunday

To guarantee integrity, Lifetime in Cache requires the use of optimistic concurrency, which locks at commit time rather than during the course of a transaction. With optimistic concurrency, an update is qualified with invariant

predicates such that the update will fail if someone else has changed the data in the database. A failed update is rolled back to maintain data integrity. For Lifetime in Cache, optimistic concurrency is essential. Data that sits in cache for long periods of time cannot hold locks but is susceptible to concurrent updates. Optimistic concurrency addresses both issues. Beans that cannot use optimistic concurrency cannot use Lifetime in Cache.

The use of Lifetime in Cache is optional and, as shown in this article, has a bigger impact for those applications with large amounts of read-only beans. Application architects should decide to use this new facility prior to application installation and deployment, based on the estimated and projected ratio of reads and updates.

Application architects and developers can use WebSphere Studio Application Developer to configure Lifetime in Cache. Alternatively, application deployment specialists can use the WAS Application Assembly Tool to configure Lifetime in Cache on an individual CMP bean basis prior to application installation. When the application architect and deployment specialist decide to use Lifetime in Cache, they also establish and configure a container-managed invalidation policy based on one of the criteria described earlier.

Flexibility Through Invalidation: Lifetime in Cache Usage Scenarios

The key feature of Lifetime in Cache is that it provides for invalidation based on one of the time-based criteria discussed earlier. With invalidation, the container automatically considers the cached entity bean state invalid when the application-configured time event occurs. After invalidation, the container will reread the entity bean's persistent state from the underlying data store on the bean's next access. Lifetime in Cache configuration allows the application architect or application deployment specialist to specify when automatic or container-managed invalidation occurs.

Unlike Lifetime in Cache, the forms of entity bean caching specified for EJB 2.0, Option A and Option B (see sidebar), don't provide for invalidation. Without container-managed invalidation there is a steady buildup of bean instances that are used for a few transactions and then never again. Without invalidation, application architects have to limit entity bean caching to those bean types that have a small number of instances and are widely used across a number of users and transactions.

The primary benefit of entity bean caching is the avoidance of subsequent rereads of an entity bean's persistent state from the underlying data store. Without container-managed and time-based invalidation the application server and EJB container have to be stopped to allow a cached entity bean's state to be reread and refreshed. With container-managed, time-based invalidation, cached beans will automatically be invalidated and then reread from the underlying data store as needed to pick up changes. Application architects and deployment specialists can configure a time-based invalidation policy consistent with the unique needs of their application.

The following scenarios illustrate the benefit of configurable invalidation:

- ***Bean Type A:*** Has several hundred instances that are frequently accessed and cached. Based on configuration options these instances are automatically invalidated at 11 p.m. each night. Between 11 p.m. and 7 a.m., there is no use of the application. However, WAS is still needed for other applications with different characteristics. During the overnight refresh cycle, 11:30 p.m. through 6 a.m., nightly batch jobs update the underlying data store for Bean Type A. The batch job runs outside of WAS, which otherwise remains available without operator intervention.
- ***Bean Type B:*** Has several thousand instances that are infrequently accessed. However, when an instance of Bean Type B is accessed, it is involved in 10 or more consecutive transactions and then typically not again. Lifetime in cache – with an invalidation policy of 120 seconds – eliminates database read overhead for 9 consecutive transactions using Bean Type B.

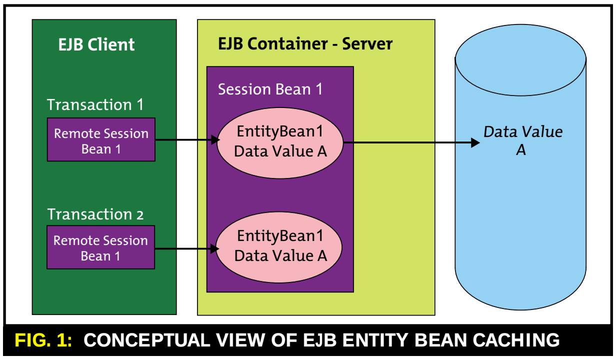


FIG. 1: CONCEPTUAL VIEW OF EJB ENTITY BEAN CACHING

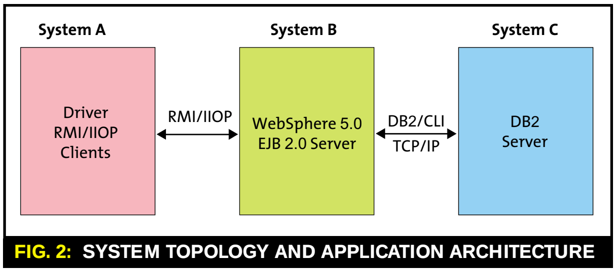


FIG. 2: SYSTEM TOPOLOGY AND APPLICATION ARCHITECTURE

- ***Bean Type C:*** Has 25 bean instances that provide control information for a major application. These 25 bean instances are used 24 hours a day Monday through Saturday and 23 hours on Sunday. For one hour on Sunday a weekly batch job outside of WAS updates the 25 beans. With an invalidation policy of Sunday, the beans are automatically invalidated just prior to the update job. WAS otherwise remains available.

Maximum Performance Benefit of Lifetime in Cache

Now I'll examine and analyze the maximum performance benefit possible using Lifetime in Cache. The next section will put this into a more realistic perspective using a typical business application.

Figure 3, based on a primitive workload, shows the maximum performance improvement achievable through Lifetime in Cache. The workload comprises multiple RMI/IIOP client threads executing successive FindByPrimaryKey operations using an even distribution of key values to prevent lock waits. In contrast, Figure 3 also shows the maximum performance improvement possible using Option A, an existing J2EE EJB Specification caching option. Figure 3 provides a further contrast by using the same performance primitive with no caching specified. (See the sidebar for more detail and contrast concerning the EJB 2.0 and EJB 1.1 Specification caching options, Option A, Option B, and Option C.)

The maximum degree of benefit shown in Figure 3 is dramatic and is attributable to the elimination of database and I/O latency, along with a related reduction in system overhead.

Figure 4 analyzes maximum performance improvement by showing the difference in where time is spent with and without caching. This difference extends to system components as well as time spent in WAS. Notice in Figure 4 that there is some – but reduced – network overhead. The residual network overhead is attributable to the RMI/IIOP clients.

Application Examples of Lifetime in Cache in Usage

Understanding the maximum potential benefit from Lifetime in Cache, we now apply it to more realistic applications to gain a better perspective of the performance benefit, which is directly related to the application's proportion of cacheable read activity. The greater the proportion of cacheable select statements to the total of all SQL statements, the greater the performance benefit.

Some applications will derive performance benefits of varying degrees. However, some applications will get no benefit. Let's look at two examples.

EXAMPLE 1 – TRADER APPLICATION

This is a stock trader–like application with five CMP entity bean types mapped to five underlying database tables.

One of the bean types, Quote, has read-only characteristics. The Quote bean accounts for 27% of the SQL statements. All of the Quote bean's SQL statements are SELECT statements. We chose to cache this bean type with a long

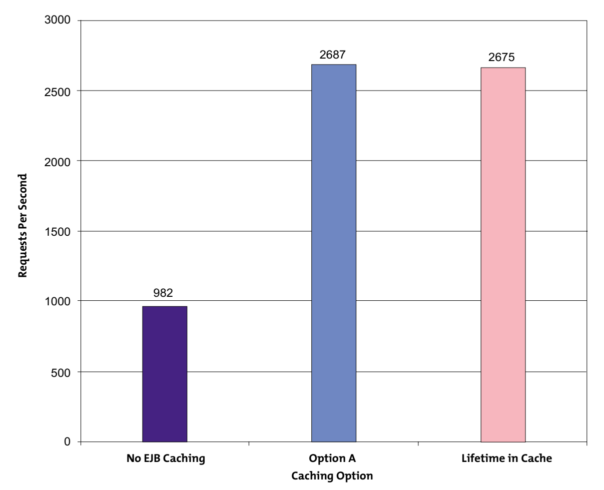


FIG. 3: MAXIMUM PERFORMANCE BENEFIT OF USING LIFETIME IN CACHE FOR FINDBYPRIMARYKEY

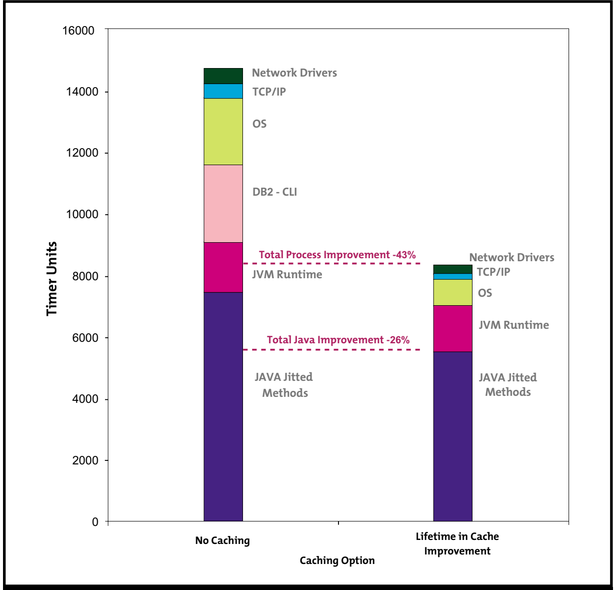


FIG. 4: WHERE TIME IS SPENT WITH AND WITHOUT CACHING

J2EE EJB 2.0 and EJB 1.1 EJB Caching Options

The J2EE EJB 2.0 and EJB 1.1 specifications provide the following caching options, which apply to both BMP and CMP Entity types. (Lifetime in Cache is for CMP beans only.)

1. **Option A:** Instances of Bean Types marked for Option A caching are loaded in the container cache one time only. Future loads from the underlying data store are avoided. Updates to the bean state are applied and written to the underlying data store under normal transaction flow. The container requires and enforces exclusive access to instances of Bean Types marked for Option A caching. There is no provision for invalidation.

2. **Option B:** Instances of Bean Types marked for Option B caching are activated and remain in the container cache. Unlike Option A, instances of Bean Types marked for Option B are loaded from the underlying data store for each transaction. As with Option A, updates to the bean state are applied and written to the underlying data store under normal transaction flow. Unlike Option A, for Option B the container does not require and does not enforce exclusive access. There is no provision for invalidation.

3. **Option C:** This is the default and provides no caching of bean instances and/or bean state between transactions.

elapsed time invalidation policy. Figure 5 shows the result, a 15% increase in throughput. These statistics detail the improvement:

- ***Rows selected:*** -25.17%
- ***Select SQL statements:*** -35.67%
- ***Total SQL statements:*** -27.21%

EXAMPLE 2 – MANUFACTURING APPLICATION

This manufacturing application has 18 CMP entity bean types mapped to 18 underlying database tables.

Four of the bean types have read-only characteristics. The result is a 6% increase in throughput. Here are the statistics around the improvement:

- ***Rows selected:*** -10%
- ***Select SQL statements:*** -10%
- ***Total SQL statements:*** -6%

Container-Managed Relationships and Read-Ahead

We now turn to the performance characteristics of container-managed relationships and the optional CMR-related Read-Ahead performance enhancement. The EJB 2.0 Specification provides CMR relationships for container-managed persistence as a programming convenience for accessing EJB 2.0 application model associations.

With CMR relationships, an association appears as a property of an entity bean. The Getter method for the association retrieves the associated entity beans that satisfy the relationship to the parent bean.

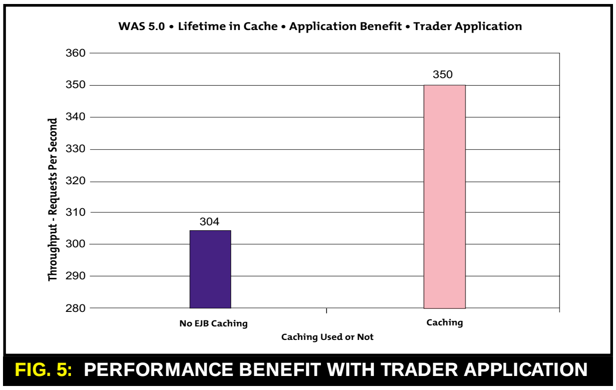


FIG. 5: PERFORMANCE BENEFIT WITH TRADER APPLICATION

Figure 6 shows two typical EJB entity bean associations that are implemented as CMR relationships. These two associations are the most common:

- One-to-one unidirectional
- One-to-many unidirectional

Read-Ahead

WAS 5.0 provides Read-Ahead as an optional capability for preloading CMR-related association entity beans during the FindByPrimary key operation for the parent bean. With Read-Ahead the associated data is retrieved from the database, but is not retrieved by the application until it accesses the Getter method for the association. Read-Ahead for preloading association-based data is set with a Read-Ahead hint. Based on the example shown in Figure 6:

- The Read-Ahead hint for Policy-Agent-Claims - FindByPrimaryKey for Policy retrieves the data for Policy and preloads the data for Agent and also the data for Claims. There is one "joined" SQL statement for all three underlying tables.

- The Read-Ahead hint for Policy-Agent - FindByPrimaryKey for Policy retrieves the data for Policy and preloads the data for Agent. The data for Claims is not retrieved until the application asks for it. There is one joined SQL statement for the Policy and Agent tables.
- The Read-Ahead hint for Policy-Claims - FindByPrimaryKey for Policy retrieves the data for Policy and preloads the data for Claims. The data for Agent is not retrieved until the application asks for it. There is one joined SQL statement for the Policy and Claims tables.
- The Default - FindByPrimaryKey for Policy retrieves the data for Policy only. The Agent and Claims data is retrieved individually when the application requests it. Without any Read-Ahead hint, the default causes the data for each association to be retrieved separately from the FindByPrimaryKey for the Policy bean.

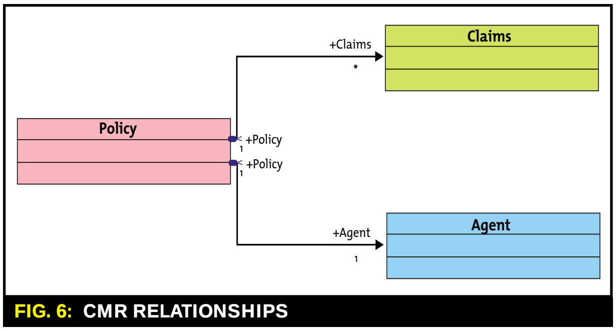


FIG. 6: CMR RELATIONSHIPS

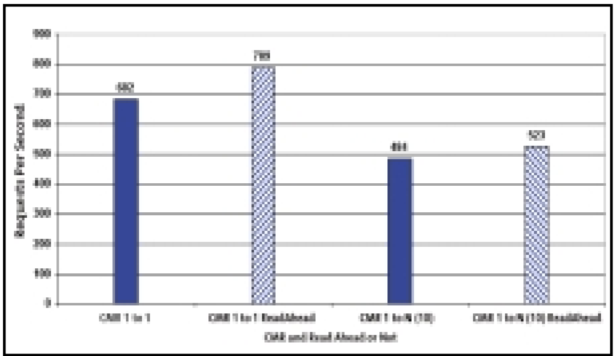


FIG. 7: COMPARISON OF EJB 2.0 CMR PERFORMANCE WITH AND WITHOUT READ-AHEAD

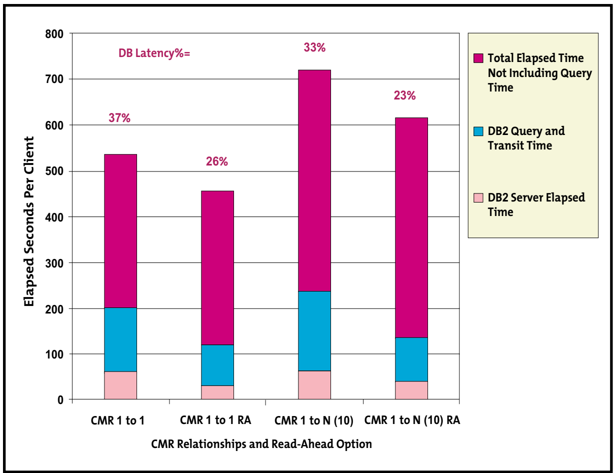


FIG. 8: ANALYSIS OF EJB 2.0 CMR PERFORMANCE WITH AND WITHOUT READ-AHEAD

Based on a primitive workload, Figure 7 compares CMR-based access with and without Read-Ahead. Figure 8 shows the analysis of the performance results in Figure 7. This analysis is based on the server round-trip time with separation of database latency. The 16% improvement for CMR one-to-one relationships is directly attributable to the reduction in database latency through use of Read-Ahead. The 8% improvement for CMR one-to-many relationships (10% in this example) is likewise directly attributable to the reduction in database latency through use of Read-Ahead.

WAS vs WAS Enterprise

WebSphere Application Server Enterprise (WAS Enterprise) v5.0 allows for one Read-Ahead hint per entity bean. Consequently, in WAS a Read-Ahead hint applies to all accesses of a given entity bean. Note that, depending on the access patterns for the entity bean, using Read-Ahead could yield worse performance than not. Consider the Policy, Agent, and Claims relationship shown in Figure 6. Suppose that in 85% of Policy accesses only Policy is needed. Consider further that in 10% of Policy accesses both Policy and Agent are required. Finally, suppose that only 5% of Policy accesses involve all three beans. In this hypothetical case, defining a Read-Ahead hint for all three beans would yield a performance drop.

WAS Enterprise provides enhanced support for less than 100% association reference patterns. WAS Enterprise allows for the definition of Read-Ahead hints on session beans as well as entity beans. Each unique business process, as represented by a session bean method, can have a Read-Ahead hint that reflects and optimizes the exact data requirements of the process.

Conclusion

Both Lifetime in Cache and Read-Ahead are important new tools in every application architect's performance optimization arsenal. Lifetime in Cache improves performance by caching CMP entity bean persistent data and reducing database latency. Lifetime in Cache has a greater performance impact for applications with a large amount of read-only bean types. By providing time-based invalidation, Lifetime in Cache gives application architects much-needed flexibility for performance optimization. Read-Ahead adds performance optimizations to augment the convenience of container-managed relationships. WAS Enterprise provides added capabilities that maximize the potential benefit of Read-Ahead.

Acknowledgments

I would like to acknowlege the following people for their help with this article: Gennaro Cuomo, IBM Distinguished Engineer and CTO of the WebSphere Technology Institute; Ruth Willenborg, manager of the WebSphere Performance team; Timo Salo, Art Jolin, Yang Lei, and Kevin Williams of the WebSphere Persistence Management team, for giving us Lifetime in Cache and Read-Ahead; and John Stecher, of the WebSphere Performance team.

Best practices for Web security planning

Untangle Your Web Security Deployment

BY PEIYIN PAI

Computer security, no matter what platform, has traditionally been viewed as a necessary evil because most users believe it hinders their freedom to do their jobs. Over the years we have come to understand the necessity of computer security for business to endure. However, computer security continues to be stereotyped as a business overhead.

In today's e-business environment, in which your financial growth depends on successful and secured business transactions, security becomes a business enabler. No longer are you just displaying static HTML pages to provide your customers with your company's product and support information – by using application server tools like WebSphere, you are now engaging in real transactions for goods or services. As a result, your customers and partners are demanding a secure environment in which to participate in e-commerce.

For security goals and controls to be successful, it is imperative that upper management support their development. The most senior executives in your corporation must be convinced that security is, indeed, an e-business enabler and can make an impact on the corporation's financial goals. With the support of upper management, an implementation team can be appointed that will formulate and deploy your corporation's security policy.

A security implementation in an enterprise environment is a major project. As with any major endeavor, you need to follow good project management guidelines to ensure a successful implementation. One of the most important actions you can take is the creation of an implementation team. The implementation of your Web security software requires a concentrated effort by the individuals assigned to the implementation team, and also requires cooperation and contribution from the areas in your organization that will be affected by the implementation.

In addition to creating an implementation team, you need to:

- Assign a project manager
- Hold regular meetings
- Establish an archive of all pertinent documentation relating to the project
- Review your corporation's security policies and procedures

Assigning an Implementation Team

Before any serious deployment of new technology can begin, it is imperative that you assemble the proper implementation team. Since WebSphere-based applications touch so many IT disciplines, it is critical that you include all of these groups in your initial planning sessions. In addition, other departments (such as marketing, order processing, manufacturing, and finance) may also need to get involved since they could ultimately be the business owners of your Web applications. Some members of this team may not be doing the actual work; however, they will definitely have a say in the deployment architecture, methodology, and time line.

Although you may have the actual vendor or a contractor run the project for your company, you should always own the implementation and have an internal team assigned to work with the deployment vendor. The following sections explain how to identify the internal members of your implementation team and define their roles and responsibilities.

Identifying Team Members

Your implementation team may consist of the individuals assigned to perform the actual implementation and representatives from each of the following affected areas:

- Security administration
- Systems software
- WebSphere applications software

- Operations
- Auditors
- Business users
- End users

Now let's take a look at the specific departments and roles in an organization that should be included in the implementation team.

Defining Roles and Responsibilities

After you have identified the organizational groups that will be involved in the planning and implementation of the deployment of your Web service security software, you need to ensure that each of their functions is clearly identified. Regardless of organizational responsibilities, the following roles should be considered and assigned to specific members of the implementation team:

- ***Project manager:*** Owns the overall project management tasks, deliverables, communications, and timetables.
- ***Security administrator:*** Responsible for the review and approval of design, architecture, and naming standards as they pertain to userIDs and resources.
- ***Operations representative:*** Responsible for the day-to-day operation of Web security applications in terms of the hardware, software, and procedures required to maintain agreed-upon service levels.
- ***Network and systems representative:*** Responsible for maintaining the connectivity of the environment in which Web security applications run.
- ***End-user liaison:*** A business person who represents the end user experience when it comes to interface decisions or user awareness issues.
- ***Business representative:*** Responsible for the policies that will affect the end user's experience with certain business Web applications as they are conducting electronic transactions for services or purchases.
- ***Management:*** The success of any project requires the constant involvement and approval of senior management at every step of the way.

Formulating a Security Policy

The initial assignment of the security implementation project team may be to develop and recommend the security policy or the documentation of security objectives for your Web environment. You may be able to use or borrow concepts from the established policies of your mainframe or distributed environments since most, if not all, Web-based applications have the same generic security requirements, such as authentication and authorization.

If the security policy or the documentation of security objectives has already been developed, the implementation team can use this document as its mandate. If these documents must be developed, the team is an ideal committee to do it since they can take into account the concerns of each affected area while developing the objectives. If each area agrees to the direction being set, which is more likely with active participation, then implementation can proceed smoothly without time-consuming discord between the areas.

After the security policy has been formulated, upper management should issue a position statement to all internal employees and appoint a security officer (or at least a security administrator). The security officer can then ensure that employees are made aware of the security policies and procedures that they must adhere to and the consequences of any security violation.

Statement of Goals

In formulating a security policy, you first need to define the actual mission or goals of this policy. For example, when establishing the security goals for your Web environment you should consider:

- The protection of corporate intellectual property
- The protection of an individual's information
- The segregation of customer, partner, vendor, or supplier information
- The security of financial transactions
- The implied trust of the Web environment you have created
- Your commitment to provide a pleasant, yet secure, Web experience

Your security policy needs to be clear and precise so that users can easily understand your corporation's security objectives. Your policies should also be easy for any type of user (intranet, extranet, or Internet user) to access without excessive searching. In other words, ignorance will not be an excuse for violating the security policy.

Ensuring User Accountability

Since security policies that are not enforced are useless, you should always strive to balance security controls with disciplinary action. It is easier to enforce your policies for internal employees, since employees are subject to corporate policies. It becomes difficult when customers, potential customers, or business partners do not adhere to your policies. Therefore, it becomes essential to have clearly defined security controls policies, enforcement methodologies, and disciplinary actions.

Key components for ensuring user accountability include:

- Ensuring that all users are made aware of your security controls policies
- Letting your users know that you have an enforcement methodology
- Letting your users know the disciplinary ramifications of any security violations
- Reviewing your policies on a regular basis to ensure that they are still meeting your business objectives
- Implementing a two-tier audit control methodology: real-time monitoring of serious security violations and postmortem audit reports

Creating an Implementation Plan

Now that you have formed a security policy that provides the foundation for success, you need to create a thorough implementation plan. Even if

ABOUT THE AUTHOR

Peiyin Pai is the *e*Trust brand marketing manager at Computer Associates. He is responsible for marketing strategy and communication, brand management, and product direction of several security products in the *e*Trust portfolio. Peiyin has worked in the IT industry since 1987 in a variety of roles, including marketing, software development, quality assurance, technical support, and project management. He has been actively managing security projects since 1996.

E-MAIL

pei-yin.pai@ca.com

your security solution will involve vendors or contractors, you need to have ownership of the entire deployment process. This will ensure that you understand the technology you are deploying and are able to make any critical business decisions when required.

Planning and scheduling the security implementation can help set the proper direction and keep the implementation on course. A well-designed implementation plan is your most important control mechanism.

An implementation plan that facilitates successful deployment contains many components. These components can include:

- Creating a good working team
- Defining the scope of the project
- Defining realistic timetables
- Defining milestones and deliverables
- Executing a small pilot environment
- Phased implementation
- Conducting product training

Typical Implementation Plan Components

The following tasks should be included as a part of a typical security implementation plan.

- ***Product training:*** Time must be allocated to allow security administrators to be trained in the use of the Web service security applications.
- ***Installation:*** Your installation of particular Web service security software depends on many site-specific factors. When planning your installation, you should consider a phased implementation to minimize any exposure.
- ***Inventory of resources and users:*** The inventory phase can be one of the most time-consuming phases of the implementation. Its duration is determined by the number of users and resources in the installation. The results of the inventory can then be used as input in a phased implementation.
- ***Definition of implementation strategy:*** Each organization may choose to approach the implementation in a different way, addressing different facilities and using different options and controls.
- ***Development of emergency and troubleshooting procedures:*** Before misuse or misconfiguration problems occur, it is critical to schedule the time to develop emergency procedures that should help minimize the time required to diagnose and resolve specific problems.
- ***Development of security maintenance procedures:*** Changes in your environment (such as new applications) may require changes to your security policies. Development of maintenance procedures should be scheduled early, in anticipation of subsequent maintenance requirements.
- ***Testing:*** A test plan should be designed to ensure that your Web service security software is implemented and functioning as desired in the installation.
- ***Security awareness programs:*** The solidity and permanence of the security implementation will depend on the support of the user community. Support will come only if the users are properly educated about the features of the security product.
- ***Ongoing assessment and evaluation:*** Ongoing assessment and evaluation programs should be developed and scheduled at regular intervals.

Defining the Scope of Work

The first task in the planning phase is to determine the scope of the deployment. Perhaps you want to control access to a subset of users or to key business-critical applications. You can deploy the security applications throughout your company's WebSphere application platforms in a very granular manner, as shown in Figure 1. Regardless of deployment scope, the same fundamental infrastructure can provide security controls at whatever level you are comfortable with.

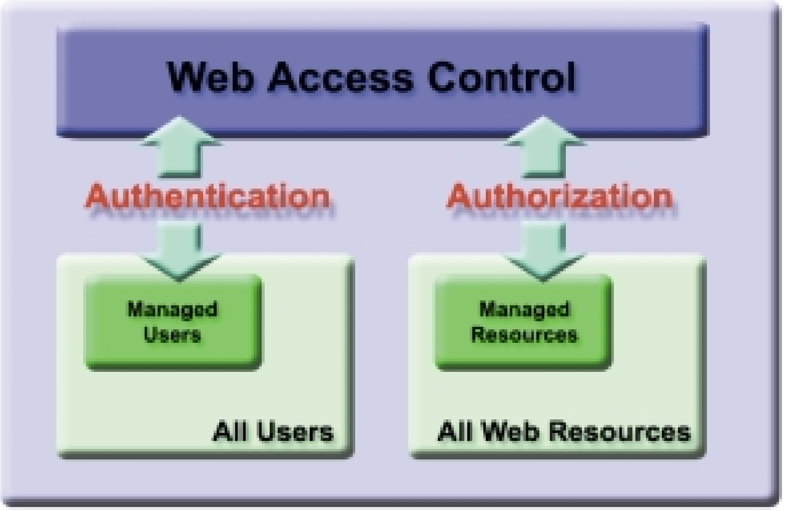


FIG. 1: WEB ACCESS CONTROL CAN REGULATE WEB ACCESS BY WAY OF USER AUTHENTICATION AND RESOURCE AUTHORIZATION

Constructing an Implementation Schedule

A complete implementation schedule consists of a task list or flowchart and a time schedule. The task list or flowchart shows all the tasks that must be accomplished to implement the Web service security applications provided by WebSphere and other application servers at your site. Developing a detailed task list or flowchart allows you to determine which tasks are dependent on each other and must be done as part of a step-by-step procedure, and which tasks are independent. By analyzing all requirements before you start the implementation, you may find that tasks targeted as part of later phases may need to be completed in an earlier phase.

The implementation team should draft a flexible time schedule. Developing realistic time estimates for the schedule depends on how well you can judge the size of the tasks. In addition, you must allow for the unknown; for example, your user base and resources may not be completely known, and hidden tasks may be uncovered as the implementation proceeds.

If at all possible, avoid setting a final implementation date until the inventory and design phases are completed. Plan to take care of the urgent requirements first, and then phase in the remainder of the organization. If careful planning and analysis are done initially, the implementation will progress smoothly and will speed up as the administrator becomes more familiar with the Web services security software, the environment, and the security administration function.

Product Training

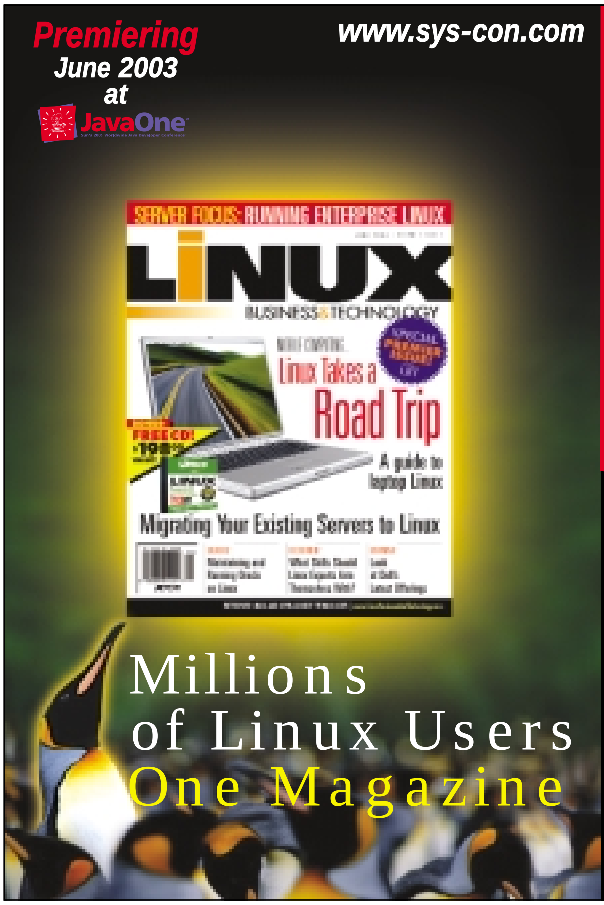
Trained individuals who understand the dynamics of Web environments and the features and functions of the product make it possible to have the most successful implementation of your Web service security. Therefore, it is important that you schedule product training as part of the overall deployment schedule. Different types of training are needed for different users – from administrators and application programmers to end users, each group of users has different training needs. The return on investment in training can be leveraged by higher operational efficiency, increased product utilization, and reduced misadministration or security risks.

Conclusion

In many organizations Web service security is a critical but often overlooked component of e-business or e-commerce project deployment. It requires the same attention and coordination that any other important component of the project does. With careful planning and concerted efforts from upper management to end users, Web service security can become a business enabler rather than an afterthought.

Computer Associates eTrust Security

Computer Associates eTrust Access Management provides policy-based dynamic security for critical business assets, ensuring end-to-end access control across multiple platform and application boundaries. It provides complete solutions for securing both Web and system resources and proactively prevents intrusions. By integrating WebSphere and other application servers into one consolidated security management layer, it enables new, secure online services today and provides a foundation for building Web services solutions in the future.



A step-by-step guide

Developing Portlets

BY BHAVESH PATEL
AND ATIT PATEL

According to IBM, "portals are the next-generation desktop." Portals are a single point of access to information for users and can provide either various types of information (horizontal portals) or detailed information on a single topic (a vertical portal). Yahoo, for example, is a horizontal portal, providing information on multiple topics. IBM has introduced WebSphere Portal Server as a foundation on which to build an enterprise-wide framework and allow services or extensions to be plugged into the Portal Server with a minimum of difficulty.

ABOUT THE AUTHOR

Bhavesh Patel, an enterprise architect with Noospherics Technologies, has over 10 years of IT experience and holds certifications from IBM, Microsoft, Compaq, and Sun. Bhavesh, who speaks and writes on WebSphere topics, has worked as a project mentor, tech lead, and WebSphere administrator for several major telecommunications and financial companies, and for the U.S. government and military.

E-MAIL

patelbj@noospherics.com

Portal provides a variety of features:

- Integration with various enterprise-wide applications
- The ability to allow access to various services as portlet applications
- Personalization and profile management
- The ability to publish and distribute information
- Security
- Federated search across multiple databases

WebSphere Portal Server comes with many ready-to-use portlets, and also provides the framework necessary to develop and deploy customized portlet applications.

This tutorial provides a step-by-step technical demonstration of how to develop a basic portlet using WebSphere Studio Application Developer and the Portal Toolkit and then deploy it on WebSphere Portal Server.

Assumptions

- WebSphere Portal Server v4.1.2 (Enable) installed on WebSphere Application Server Advanced Edition, v4.0.2
- Portal Server with all prerequisites and up and running.
- You are able to get the portal server by typing http://yourhost.yourdomain.com/wps/portal. Make sure, yourhost.yourdomain.com is defined in your WINNT/System32/drivers/etc/hosts file.
- You have administrator privileges on the server. (You should be able to log on to your Portal Server using the wpsadmin username).

Developing a Basic Portlet

We will develop a basic portlet using the IBM Portal Toolkit on WebSphere Studio Application Developer installed on Windows 2000. This is a three-step process that we will describe here.

- Install and configure the Portal Toolkit.
- Develop a basic portlet using Portal Toolkit in WebSphere Studio Application Developer.
- Deploy the portlet and test it on WebSphere Portal Server v4.1.2.

Installing the Portal Toolkit

Download the free Portal Toolkit plug-in available from IBM from www-3.ibm.com/software/ info1/ websphere/index.jsp?tab=products/portaltoolkit&S_TACT=10. The toolkit allows you to create a portlet development environment for WSAD. Using the Portal Toolkit, you can create portlet application projects wherein you can develop your portlets.

Follow these steps to install the Portal Toolkit.

- Once you have downloaded the Portal Toolkit executable, PortalTookit4.1.exe, run the self-extracting executable and extract it into a temporary directory. From the temporary directory, run install.bat.
- Click Next in the Portal Toolkit - InstallShield Wizard screen. You will get a warning asking you to stop WebSphere Studio Application Developer if it is running. Quit WSAD if it is running.
- Select the "I accept the license…" radio button and click Next. (You don't have another choice, do you!)
- The wizard will ask you to confirm the installation directory for WSAD. You can leave the default "C:\Program Files\ IBM\Application Developer" if that is your installation folder. Otherwise click the Browse button and select your desired installation directory.
- Click Next when the InstallShield wizard informs you of the size of the toolkit that will be installed. You should get a message confirming

the successful installation of the toolkit. Click Next, and click the Finish button to complete the installation of the Portal Toolkit.

Developing the Portlet Using WSAD/Portal Toolkit

When you restart WSAD after installing the toolkit, you should now have two new wizards added to your WSAD configuration (see Figure 1).

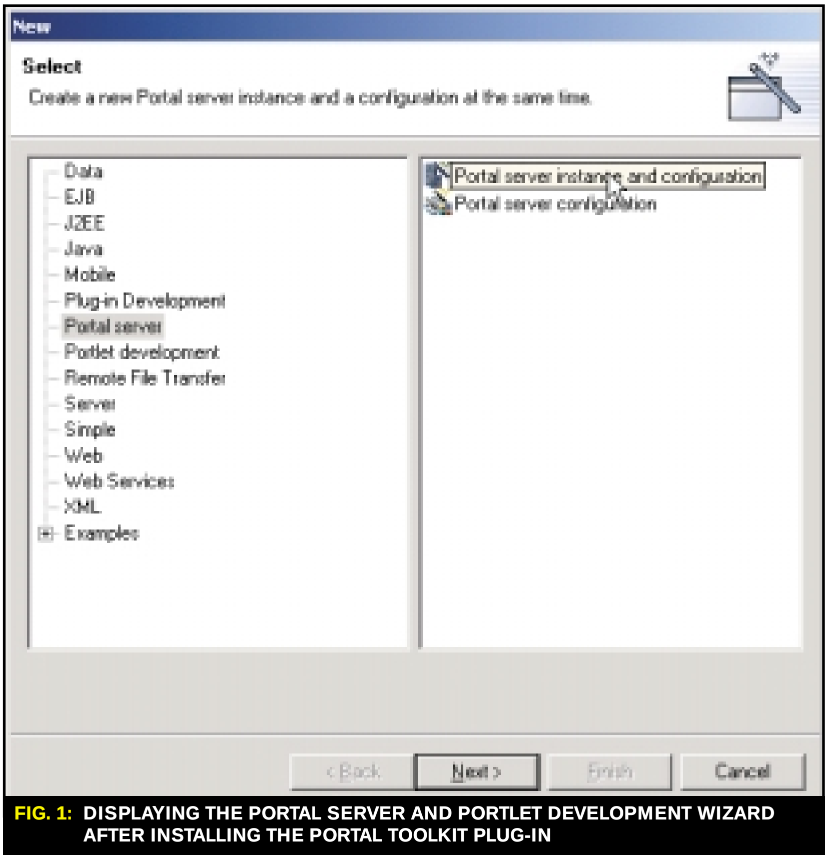
- The Portal Server wizard, which includes the Portal Server Instance and Configuration Project and the Portal Server Configuration Project
- The Portlet Development wizard, which has a Portlet Application Project

We will start by creating a portlet application project, called MyPortletProject, in which we will develop our portlet. Follow these steps to create the portlet application project.

- From the WSAD menu bar, select File–>New–>Other.
- You will be prompted with a select screen that allows you to choose a wizard. Select the Portlet Development wizard in the left panel, select the Portlet Application Project in the right panel, and click Next.
- In the Create a Portlet Project window you will be asked to "Define a Project." Enter the values shown in Table 1.
- In the Create a Portlet Project window you will be asked to make a "Portlet Selection." Select the radio button for a Basic portlet and click Next.
- Now you will be asked to enter "Basic Portlet Parameters." Enter the values shown in Table 2.
- Click on the HTML checkbox. You can click others if your portlet needs to support devices other than the browser as a client, e.g., select WML to support cell phones as a client for your portlet. Your screen should look like the one in Figure 2. Click on the Finish button.

Project Name	MyPortletProject
Enterprise Application Project Name	MyPortletApplication
Context Root	MyPortletProject (the default)

TABLE 1: DEFINING A PROJECT



Create a new Portal server instance and a configuration at the same time.

FIG. 1: DISPLAYING THE PORTAL SERVER AND PORTLET DEVELOPMENT WIZARD AFTER INSTALLING THE PORTAL TOOLKIT PLUG-IN

WSAD will open up the Portlet Perspective (this may take a while) and you will see your portlet application project and its corresponding JSP files, portlet.xml, and the rest of the directory structure (see Figure 3).

You will find the MyPortlet.java and MyPortletBean.java files in the com.myportal package in the "source" folder in your project. MyPortlet.java is a class that extends the PortletAdapter class and contains the doView(), doEdit(), and doHelp() methods that are called when the Portlet Application switches to the corresponding modes. MyPortletBean.java contains the getters and setters for the variables used to show the values on the portlet application view page.

- ***View.jsp:*** Page for the view mode
- ***Edit.jsp:*** Page for the edit mode
- ***Help.jsp:*** Page for the help mode
- ***Portlet.xml:*** Portlet deployment descriptor

Note that to ensure that the user is able to use the edit and help modes (only view is enabled by default), you have to take the following steps.

- Open portlet.xml in your WSAD project in navigation view and double-click the portlet.xml file.
- Expand the Portlet Application folder by clicking on the + sign.
- Select Portlet_1. In the Supports section, you will see Markup, View, Configure, Edit, and Help.
- The Edit and Help options are selected to None, which you will change to Fragment, available from the dropdown menu.
- Click on the close button and Select Yes to save the modified portlet.xml.

A Note to the User

We have modified the source code for the MyPortlet.java, MyPortletBean.java, view.jsp, edit.jsp, help.jsp, and the portlet.xml files.

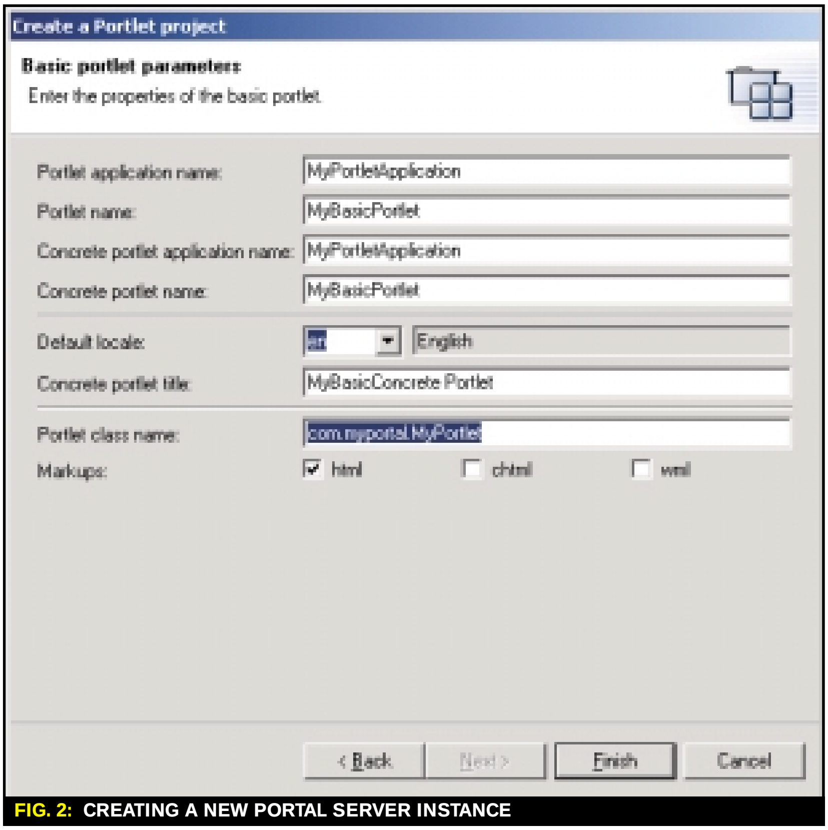
The source code for this article can be downloaded from www.sys-con.com/websphere/sourcec.cfm. We have also provided a WAR file that contains the complete WAR format export of the portlet application project along with source code, for readers who do not

ABOUT THE AUTHOR

Atit Patel is an enterprise developer for Noospherics Technologies with more than six years of IT experience. Atit is a Sun Certified Java Programmer who has been involved with WebSphere Portal right from its release.

E-MAIL

atit@noospherics.com



Create a Portlet project

Basic portlet parameters

Enter the properties of the basic portlet.

Portlet application name: MyPortletApplication

Portlet name: MyBasicPortlet

Concrete portlet application name: MyPortletApplication

Concrete portlet name: MyBasicPortlet

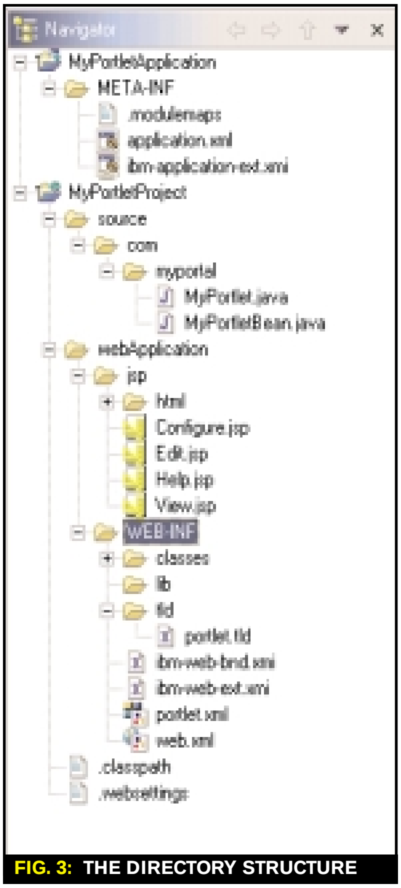
Default locale: en English

Concrete portlet title: MyBasicConcrete Portlet

Portlet class name: com.myportal.MyPortlet

Markups: ☑ html ☐ chtml ☐ wml

FIG. 2: CREATING A NEW PORTAL SERVER INSTANCE



Navigator

```
MyPortletApplication
    META-INF
        .modulemaps
        application.xml
        ibm-application-ext.xmi
MyPortletProject
    source
        com
            myportal
                MyPortlet.java
                MyPortletBean.java
    webApplication
        jsp
            html
            Configure.jsp
            Edit.jsp
            Help.jsp
            View.jsp
        WEB-INF
            classes
            lib
            tld
                portlet.tld
            ibm-web-bnd.xmi
            ibm-web-ext.xmi
            portlet.xml
            web.xml
    .classpath
    .websettings
```

FIG. 3: THE DIRECTORY STRUCTURE

want to install the Portal Toolkit but wish to work through part of the tutorial with an existing portlet application.

This application demonstrates a basic portlet, and shows how the view, edit, and help modes of a given portlet can be used. This is a very basic application in which the view.jsp reads certain values already set in the session. When you click the Edit icon on the view.jsp page you are taken to edit.jsp, which gets you into edit mode. We allow you to modify/edit the values in edit mode and when you click done you are returned back to the home page, and the view page is displayed using the modified set of values. In order to avoid complexity, we use a session to act as a lightweight database to store the values for our application.

Note: We have modified the default code that gets generated when you create the portlet. Hence, when you follow the steps in this tutorial and compare the default generated code for the aforementioned three JSPs and two Java files, you will see a slight difference. We did this to give a cleaner view and to make things easier for developers using this tutorial.

Once we have made the code changes to these files, you can export the WAR file for the application project using the following steps.

- Select File–>Menu–>Export.
- Select WAR and click Next.
- Select MyPortletProject, when asked to select the resources to export.
- Define where you want to export the WAR file to, including the complete path and WAR file name. For example, c:\Portal\MyPortlet.war.

Deploy and Test the Portlet

Now we are ready to deploy our portlet application. Deployment is divided into three steps.

- Install the portlet WAR file.
- Place the portlet in a page.
- Activate the page and test the portlet.

INSTALLING THE PORTLET WAR FILE

To install the portlet WAR file, go to the portal server home page by typing http://yourhostname.yourdomain.com/wps/portal. Log in as wpsadmin and supply the password. The portal home page should be similar to Figure 4.

Select Portal Administration from the dropdown menu at the top left corner of the page. The dropdown menu will offer three options: Home, Work with pages, and Portal Administration. The default selection is Home.

You should get a page asking you to specify the location of the WAR file. If you don't get this page, make sure you have selected the Portlets–>Install Portlets tab.

- Click the Browse button, select the MyPorlet.war file, and click the Next button. You should get a message saying, "The following portlets will be installed:".
- Click the Install button. You should get a message saying, "Portlets successfully installed."
- Click the Next button, if you have other WARs to install.

PLACING THE PORTLET IN A PAGE

- Go back to the dropdown menu and select "Work with Pages."

- Under Edit Layout and Content, select Home from the Place dropdown menu. You should get a new page asking you to “Please select the page you wish to modify.”
- In the Page dropdown menu, select the Welcome page. For this tutorial, we are adding this portlet application to the Welcome page. You can create your own pages and put your application wherever you want. But for this tutorial stick with our instructions.
- You should be prompted with a page that allows you to do a “Get Portlets.” Click on the Get Portlets Icon. Notice the warning on the page, “This page will be deactivated when you begin working on it. Please remember to activate when you are done.”
- You will be presented with the Get Portlets page, where you will find the portlet you want to install and add it to the portlet list. Do this by clicking on the radio button “Search for Portlets.”
- Type “my*” in the search box and click on the Go icon. You will see MyBasicPortlet in the Search Results Table, displayed on the same page (see Figure 4).
- Click the “plus” icon near the portlet name so that it gets added into the Portlet List and click the OK icon. You will be presented with a page where you can click the icon in left column and the portlet will be added into that column on your page.

ACTIVATE THE PAGE AND TEST THE PORTLET

- Click the Activate button. The page will be activated and the control will return to the same page.
- Just above the Edit and Layout tab, there is a dropdown menu. From the dropdown menu, select Home to load the Home page – this time with the new portlet application. You should see a window similar to that shown in Figure 5. If you still see the old home page without the new portlet application, log out and log in again as wpsadmin, and your portlet application should now be available.
- Click on the Edit icon in the title bar of MyBasicConcretePortlet.

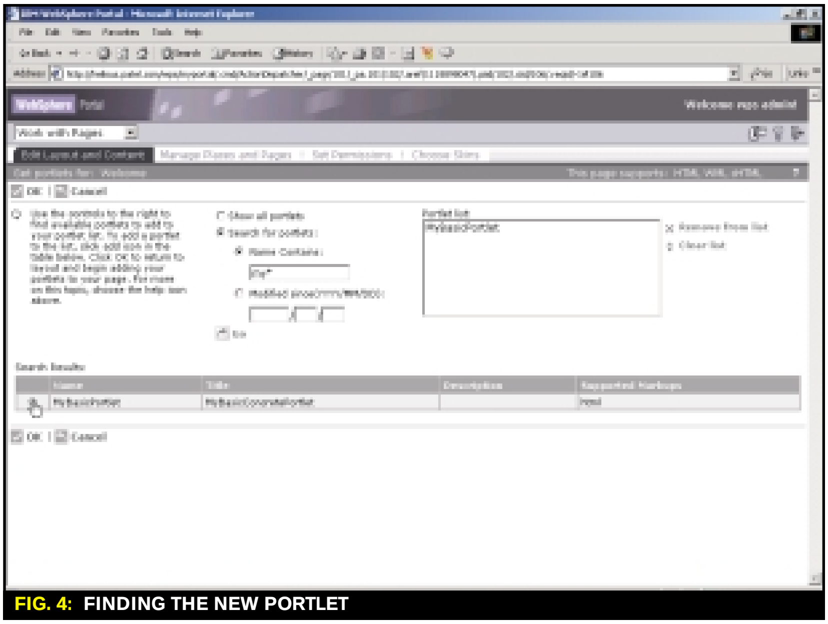
The Edit page will display with the page’s values in edit mode. You can modify the name and other values such as address, etc., and click the Done button. The control will return back to the Home–> Welcome page with the view page of the MyBasicConcretePortlet displaying the updated values.

Conclusion

Congratulations! You have successfully developed and deployed a Portal application on WebSphere Portal v4.1.2.

Portlet application name	MyPortletApplication
Portlet name	MyBasicPortlet
Concrete Portlet application name	MyPortletApplication
Concrete Portlet name	MyPortletApplication
Default locale	en (English)
Concrete Portlet title	MyBasicConcrete-Portlet
Concrete class name	com.myportal.MyPortlet
Markups	HTML

TABLE 2: BASIC PORTLET PARAMETERS



IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
WebSphere Portal
Welcome wps admin
Work with Pages
Edit Layout and Content | Manage Places and Pages | Set Permissions | Choose Skins
OK | Cancel
Go

Get portlets for: Welcome

This page supports: HTML, WML, cHTML

Use the portlets to the right to find available portlets to add to your portlet list. To add a portlet to the list, click add icon in the table below. Click OK to return to layout and begin adding your portlets to your page. For more on this topic, choose the help icon above.

○ Show all portlets
● Search for portlets:
● Name Contains: my*
○ Modified since(mm/dd/yyyy):

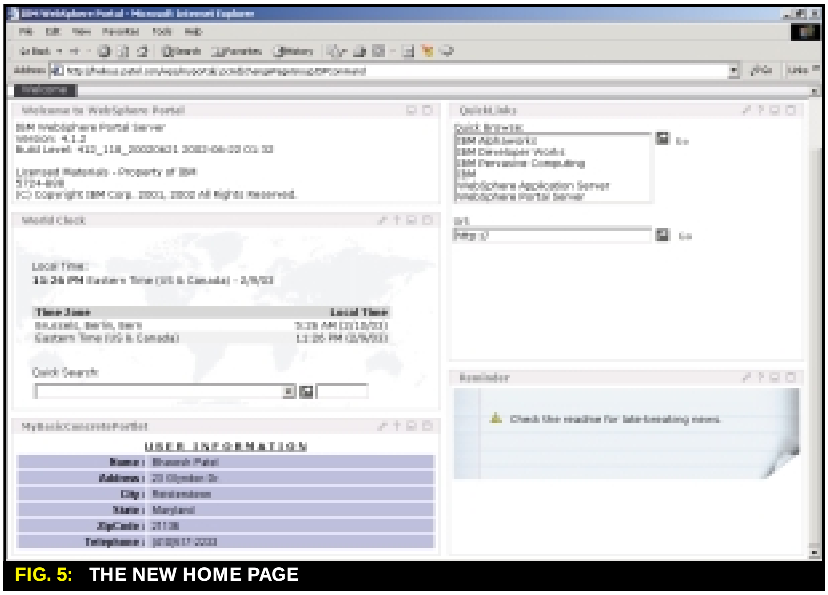
Portlet list: MyBasicPortlet

✕ Remove from list
Clear list

Search Results

Name	Title	Description	Supported Markups
MyBasicPortlet	MyBasicConcretePortlet		html

FIG. 4: FINDING THE NEW PORTLET



IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Welcome
Go
Go

Welcome to WebSphere Portal

IBM WebSphere Portal Server
Version: 4.1.2
Build Level: 412_118_20020821 2002-08-22 01:32

Licensed Materials - Property of IBM
5724-B08
(C) Copyright IBM Corp. 2001, 2002 All Rights Reserved.

World Clock

Local Time:
10:26 PM Eastern Time (US & Canada) - 2/9/03

Time Zone	Local Time
Brussels, Berlin, Bern	5:26 AM (2/10/03)
Eastern Time (US & Canada)	11:26 PM (2/9/03)

Quick Search

Quick Links

Quick Browse:
IBM AlphaWorks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Portal Server

Url:
http://

Reminder

Check the readme for late-breaking news.

MyBasicConcretePortlet

USER INFORMATION

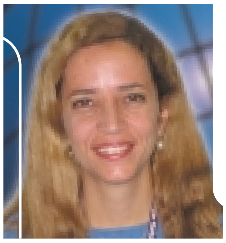
Name:	Bhavesh Patel
Address:	23 Glyndon Dr
City:	Reisterstown
State:	Maryland
ZipCode:	21136
Telephone:	(410)517-2233

FIG. 5: THE NEW HOME PAGE

JAX-RPC and JSR-109 to become part of J2EE standard

Web Services Standards

BY DENISE GABARDO
AND ANDRÉ TOST



ABOUT THE AUTHOR

Denise Gabardo has been working in application development since 1985 and has been a member of the IBM Software Group since 1996. Her main responsibility is to provide technical sales support to customers using WebSphere Application Server and WebSphere application development tools such as WebSphere Studio Application Developer and VisualAge for Java.

E-MAIL
deniseg@br.ibm.com

If you follow the latest trends in the software industry, you will have noticed that Web services technology is getting a lot of attention. While it is not a completely new thing anymore, more companies are getting serious about Web services today and putting solutions into production that provide and/or consume Web services interfaces. One crucial aspect of this is standardization. The promise of Web services technology is to allow you to connect applications that were developed on different platforms and in different programming languages. This can only work if vendors can agree on common standards. And a lot of standardization has taken place in the XML space with the creation of specifications such as SOAP, WSDL, and UDDI.

However, until recently there was no common approach to implementing Web services technology in Java. This has changed with the advent of two new standard API definitions that will go into the next J2EE specification. Both were handled in the Java Community Process as JSRs (Java Specification Requests), namely JSR-101 (JAX-RPC) and JSR-109 (Enterprise Web Services). In short, JSR-101 defines the mapping of WSDL to Java and vice versa. It also defines a client API to invoke a remote Web service and a runtime environment on the server to host a Web service. JSR-109 conceptually enhances JSR-101 by defining the packaging of Web services into standard J2EE modules, including a new deployment descriptor. It also defines Web services that are implemented as Enterprise JavaBeans.

IBM provides an early implementation for both JSR-101 and JSR-109 in the Web Services Technology Preview for WebSphere Application Server 5, giving you a head start on using the new APIs. We will cover this technology preview in detail in a later article.

In this article, we will describe the key concepts of the new standards. This will help you understand how you can integrate Web services technology into your J2EE-based application development efforts, and which new APIs and deployment descriptor formats are defined.

We will assume that you are familiar with the basic Web services technologies such as SOAP and WSDL. (See the Resources section at the end for pointers to more material on those topics.)

JAX-RPC (JSR-101)

JAX-RPC stands for Java API for XML-based RPC. As we mentioned earlier, it defines how Java classes can be described in a WSDL document and how an existing WSDL document can be implemented in Java. Which way you go depends on your development approach. In the "top-down" approach, you start with an existing interface definition that is described in WSDL and derive Java classes from that. In the "bottom-up" approach, you start developing your Java implementation and then generate a WSDL description from that.

WSDL/JAVA MAPPING

We cannot cover all of the rules that apply for the mapping between WSDL and Java, but here are some of the core ones:

- Each WSDL <portType> element is mapped to a Java interface. This interface must extend java.rmi.Remote and is called the service endpoint interface. Its package name can be derived from the namespace of the <portType> element and vice versa.
- Each <operation> within the <portType> is mapped to a Java method in the service endpoint interface. Every method must throw a java.rmi.RemoteException.
- The <message> elements of the <operation> are mapped to parameters and return types of the methods of the service end-

point interface. The mapping works like this (note that in actual usage this mapping can be quite complex, and we only list the basic concepts here):

–Basic Java types are mapped to basic XML Schema types. For example, a java.lang. String is mapped to an xsd:string type.

–Nonbasic Java types (i.e., regular Java classes) must implement java.io.Serializable. They are mapped to XML structures described in the form of XML Schema complex types. The process of turning a Java object into an XML document, and vice versa, is handled by serializers and deserializers.

–Support for XML attributes is optional; enumerations are mapped to a new Java class in a particular way; and arrays are handled as described in the SOAP 1.1 specification. It is beyond the scope of this article to describe all of this in detail.

The second article in this series will describe tools offered by the WAS 5 Web Services Technology Preview that can generate the Java interface from an existing WSDL file, and generate WSDL from an existing Java interface – so you don't need to be an expert on all of these rules.

Client-Side Service Invocation

On top of the Java-WSDL mapping, the JAX-RPC specification defines a client interface for the invocation of a Web service. The most important interface here is the javax.xml.rpc. Service interface. It defines methods to retrieve a local stub to a Web service, which then forwards each request to the actual service implementation. These stub classes are also generated by tools, so you don't have to worry too much about their internals. There is also a way to invoke a service completely dynamically, without a generated stub.

Server-Side Runtime Model

On the server side, it is assumed that the service implementation class exists in a servlet container. But the service implementation class is not the servlet that handles the SOAP request; that servlet is provided by the JAX-RPC implementation. The service implementation class will be a JavaBean that implements the service endpoint interface we mentioned earlier.

The model for EJBs is defined in JSR-109, which we will explain a little further on.

In case you are wondering now about the supported protocols for the actual communication between a service requester and a service provider, JAX-RPC is meant to be protocol-neutral but requires support for SOAP over HTTP. In other words, transport of SOAP messages over JMS is not (yet) defined.

By the way, the JAX-RPC specification requires that an implementation support both RPC-style invocations with encoded SOAP messages, and document-style invocations with literal encoding. Other combinations are optional. The Web Services Interoperability Organization (www.WS-I.org) currently leans toward the use of literal encoding (some may describe this as no encoding at all), so you can expect literal encoding to become dominant (plus, it was recently decided that the WS-I Basic Profile will become part of the J2EE 1.4 specification).

JSR-109 (Enterprise Web Services)

In order to make Web services usable within a J2EE environment, the JAX-RPC specification is not enough. For example, it does not define how a Web service is packaged and deployed, or how an EJB can be exposed through a Web service interface. This is where JSR-109, describing "Enterprise Web Services," comes in. It is basically filling the gaps that the JAX-RPC specification left open for use of Web services in a J2EE application server. And again, we will just cover the main concepts here.

Client Service Invocation

Besides using the javax.xml.rpc. Service interface from within a standard Java application, you can also use it from within a J2EE client container. In this case, the JSR-109 specification defines how a service reference can be obtained from a JNDI naming context, similar to the way an EJB Home reference is retrieved.

The specification also defines a client Web services deployment descriptor, Webservicesclient.xml, for this purpose. It consists mainly of a service reference that contains the name and location of the WSDL file for the invoked service, its service endpoint interface, and the name under which the service can be found in JNDI. And like many of the other artifacts that the JSR-101 and JSR-109 specifications define, it is normally generated by tools, so you don't have to deal with creating it manually.

Server Service Deployment

In addition to the servlet-based runtime model described in the JAX-RPC specification, JSR-109 also allows a service to be implemented by means of a stateless session EJB. By the way, this indeed means that entity EJBs or stateful session EJBs cannot be exposed as Web services. The EJB's remote interface in this case acts as the service endpoint interface. If you create a separate service endpoint interface, it must contain a subset of the methods in the remote interface.

The application server must ensure during deployment that an incoming SOAP over HTTP request is routed to the appropriate EJB. If the service implementation is a JavaBean, it must also make sure that incoming requests are handled properly. To create the needed artifacts for this, a Web service deployment descriptor is defined that contains information about the service and how it should be deployed.

But before we take a closer look at the deployment descriptor, let us focus for a moment on the packaging. Web services are packaged in their respective J2EE modules, i.e., an EJB goes into an EJB module, and a JavaBean goes into a Web module (or a dependent Java JAR file). Both are packaged into an enterprise archive (EAR) file for deployment into the application server.

The Web services deployment descriptor, webservices.xml, goes into either the META-INF directory of the EJB module, or the WEB-INF directory of the Web module. This deployment descriptor contains a description of each service that is to be deployed. It contains a pointer to the WSDL document for the service, as well as a pointer to a particular <port> within this document. Finally, it defines the service endpoint interface and has a link to the element (EJB or JavaBean) that implements the actual service. This is done via a link to either the web.xml deployment descriptor (in case the service is implemented as a bean) or the ejb-jar.xml deployment descriptor (in case it is a stateless session EJB).

When the application that includes the service is installed on the application server, the server generates all of the items necessary to make the service accessible for clients. One important aspect is that it updates the address of the service in the WSDL file (contained in the <port> element) to the actual endpoint where the service resides. The resulting updated WSDL document can then be published to a file location or even to a UDDI registry.

Outlook

At this point you may have noticed that one important aspect of Web services technology is missing: asynchronous, message-based communication as a means to implement and access a Web service. Speaking in J2EE terms, it has not yet been defined how JMS can be leveraged as a transport protocol API in Web services scenarios. Some support for this exists today in various application servers, but nothing has been standardized yet. Thus, it is probably fair to assume that the next set of Web services–related J2EE specifications will address that. Moreover, we can expect that the WS-Security standard will be mapped into the J2EE security model.

Summary

With the advent of standard APIs for Web services, this technology finally becomes a core part of the J2EE specification. This will help developers create portable and interoperable Web service providers and consumers across a variety of application servers. JAX-RPC defines the basic mapping of Java to and from WSDL, including some basic APIs for consuming Web services, while JSR-109 contains the definition for packaging Web services and making them a standard component of every J2EE-compliant application server.

Resources

- *The JAX-RPC home page:* http://java.sun.com/xml/jaxrpc
- *The JSR-109 page:* http://jcp.org/en/jsr/detail?id=109
- *The WAS 5 Web Services Technology Preview:* www7b.boulder.ibm.com/ wsdd/downloads/web_services.html
- "WebSphere Version 5 Web Services Handbook," a recently published draft redbook, discusses all aspects of Web services in WebSphere 5: http://publib-b.boulder.ibm.com/ Redbooks.nsf/RedpieceAbstracts/sg246891.html?Open
- High, R., et al. (2003). *Professional IBM WebSphere 5.0 Application Server.* Wrox: www.wrox.com/books/1861005814.htm
- *The IBM developerWorks Web services zone:* www.ibm.com/developerworks/webservices

ABOUT THE AUTHOR

André Tost works as a solution architect in the WebSphere Business Development group, where he helps IBM's Strategic Alliance Partners integrate their applications with WebSphere. His special focus is on Web services technology throughout the WebSphere product family. Before his current assignment, he spent 10 years in various development and architecture roles in IBM software development, most recently for WebSphere Business Components. Originally from Germany, he now lives and works in Rochester, MN.

E-MAIL

atost@us.ibm.com

From Borland Software Corporation

Borland Together Edition for WebSphere Studio

REVIEWED BY JAY JOHNSON

I have long been a TogetherSoft fan, having used TCC (TogetherSoft Control Center) for several years. It was a great development tool, effortlessly keeping code in sync with diagrams and providing the best reverse-engineering features in the business. TCC had lots of other goodies, such as modeling in color, metrics, and support for patterns, making it more fun to use than Rational's modeling tools. Sadly, TCC was always out of sync with IBM's WebSphere Application Server. For example, it never fully supported EJB deployment to WAS 4.0. This made J2EE development for WebSphere somewhat of a challenge for TCC-based developers, forcing some of them to switch to other IDEs, such as Borland's JBuilder Enterprise, which shipped with a built-in UML modeler from Rational.

Product Info:

COMPANY INFO:
Borland Software Corporation
100 Enterprise Way
Scotts Valley, CA
95066-3246
E-mail: sales@togethersoft.com
Web: www.togethersoft.com
Telephone: 831 431-1000

SYSTEM REQUIREMENTS:
WSAD version 5.0

I've been using IBM's flagship IDE, WebSphere Studio Application Developer (WSAD) for about a year, and I'm impressed with how smoothly J2EE applications can be deployed to WebSphere from WSAD. WSAD is based on IBM's open-source Eclipse IDE, and provides developers with an impressive arsenal of features. The best feature may be that WSAD ships with an integrated version of WebSphere Single-Server Edition for testing.

Meanwhile, Borland bought TogetherSoft, and partnered with IBM to make TCC a platform for WebSphere development. Instead of plugging deployment tools into TCC, Borland/TogetherSoft decided on a better solution: plug TCC into WSAD and call it Borland Together Edition for WebSphere Studio (TWS). Since IBM owns WSAD, it will (hopefully) stay in sync with the latest versions of WebSphere. Now, Borland and IBM appear to be both partners and competitors.

The first thing I noticed about Together Edition was how tightly it is integrated into WSAD. To create a class diagram for existing code, it was necessary only to open a UML explorer perspective and select a project. To deploy EJBs or any J2EE component, all you need to do is deploy from the appropriate diagram and WSAD takes over, taking care of all the AAT (application assembly tool) details.

A favorite TCC feature of many developers was the ability to autogenerate sequence and collaboration diagrams from any method defined in a Java class. This feature is not available in the current version of Together Edition, but will be available in a future release. Two fundamental features included in TWS are Peter Coad's *Java Modeling In Color* paradigm, and an effortless, nearly instantaneous transformation of code into class diagrams and back again. These are the features that built much of TogetherSoft's reputation in the marketplace.

Assuming WSAD 5.0 is already installed, once you install Together Edition you will get a new item in your Windows Programs menu called "Together WSE." When you start the program, you immediately have the ability to look at any Java code as a UML class diagram just by selecting the Together WSE Modeling perspective, then selecting the UML Explorer view. From this view you can double-click on the default diagram (or any other diagram) within any package in a project. Figure 1 shows the result of selecting the modeling perspective and opening the auction example that ships with WSAD 5.0. Note the picture-in-picture overview; this is a handy feature.

Together Edition supports all of the UML 1.4 diagrams. Additions to the UML standard include shortcuts (the same modeling element in different views) and hyperlinks (links from one modeling element to another). TogetherSoft provides eight kinds of UML diagrams in Together Edition:

- Class/package
- Use case
- Activity
- Component
- Deployment

ABOUT THE AUTHOR

Jay Johnson is an independent J2EE architect.

E-MAIL
aguy4ejbs@yahoo.com

- Sequence
- Collaboration
- State

Quality Assurance Tools

Together Edition provides audits and metrics to help you find common errors and overly complex areas of the code base. Audits range from the coding and style audits that have always been a vital part of TogetherSoft projects to technology-specific audits such as those involved with EJB development.

AUDITS

Together Edition provides a wide variety of audits, ranging from design issues to naming conventions. The process of running audits begins with selecting the specific rules to which your source code should conform. Together Edition generates an audit report that displays only the violations of those rules. You can examine each violation and decide whether to correct the source code.

PROBLEM-DETECTION AUDITS

For most violations, the audit report generated by Together Edition indicates the line of code that causes the violation. For some audits, however, such a line number is inappropriate. These are called problem-detection audits. An example is the Misplaced Class audit, in which the package of the class is deemed inappropriate because of the dependency between the class and a different package. For problem-detection audits, TWS uses one or more detection metrics to analyze the code to determine audit violations. TWS audit reports show problem-detection audits along with the line-oriented audits.

METRICS

Together Edition provides a wide variety of metrics ranging from lines of code to comment ratio. When you run metrics in TWS, you first select which metrics are important for your project. You can use the metrics results that TWS generates to determine which code needs to be redesigned, or you can use the results to create reports and compare the overall impact of changes in a project.

Together Edition provides documentation support for your Java projects, supporting Javadoc functionality and more. Output is HTML and can be displayed in the Help browser or in your preferred Web browser.

Still Together

Together Edition is essentially TCC integrated into WSAD, with the following differences:

- Together Edition only generates and reverses Java code; TCC works with a variety of languages.
- Together Edition does not support automated sequence/collaboration diagram generation from existing code, as does TCC. This will be corrected in a future version.
- Together Edition supports not only using and creating patterns, but matching patterns to diagrams. This goes beyond TCC's capabilities.
- Models can be moved to and from Rational Rose by means of the XMI import/export facility included in Together Edition.

Patterns

Together Edition allows the creation, recognition, and use of design patterns above and beyond those provided by WSAD. The pattern recognizer will detect design patterns in new and existing Java source code. Templates allow new patterns to be created as users see the need to add them for design and domain-specific extensions (see Figure 2).

You can apply preinstalled patterns to new or existing code. Together Edition detects such pattern in your code through its pattern recognizer. You can set Pattern Recognition to run automatically during each build or manually between builds. You can manage patterns, apply them, and save them. You can also apply patterns directly in your model using the Diagram view context menus or the UML Explorer.

WebSphere DEVELOPER'S JOURNAL

Advertiser Index...

COMPANY	URL	PHONE	PG
GLOBAL KNOWLEDGE	WWW.GLOBALKNOWLEDGE.COM	800-COURSES	39
IBM	WWW.IBM.COM/WEBSPHERE/OPENTOOLS	800-426-4968	5
LINUX BUSINESS & TECHNOLOGY	WWW.SYS-CON.COM/LINUX	888-303-5282	31
MACROMEDIA	WWW.MACROMEDIA.COM	415-252-2000	51
MQSOFTWARE	WWW.MQSOFTWARE.COM/GUIDE	952-345-8720	6,7
PERFORMANT	WWW.PERFORMANT.COM/WEBSPHERE1	866-773-6268 x0	25
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	17
REPORTINGENGINES	WWW.REPORTINGENGINES.COM/DOWNLOAD/WSDJ1.JSP	888-884-8665	21
SITRAKA (now part of Quest Software)	WWW.SITRAKA.COM/JCLASS/WDJ	800-663-4723	27
SITRAKA (now part of Quest Software)	WWW.SITRAKA.COM/PERFORMANCE/WDJ	800-663-4723	11
SYS-CON E-NEWSLETTERS	WWW.SYS-CON.COM	201-802-3020	37
SYS-CON PUBLICATIONS	WWW.SYS-CON.COM/SUBOFFER.CFM	888-303-5282	43
TRILOG GROUP	WWW.FLOWBUILDER.COM	800-818-2199	52
VERSATA	WWW.VERSATA.COM/BUSINESSLOGICDESIGNER	800-984-7638	2
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	3

WebSphere DEVELOPER'S JOURNAL

Coming Next Month...

INTERVIEW:

The Evolving Role of WebSphere: Part 2 of a conversation with the WebSphere marketing team

BY JACK MARTIN

WEB SERVICES:

Creating Web Services from Stored Procedures Using WebSphere Studio

BY JOAN HAGGARTY AND CHRISTINA LAU

PERFORMANCE:

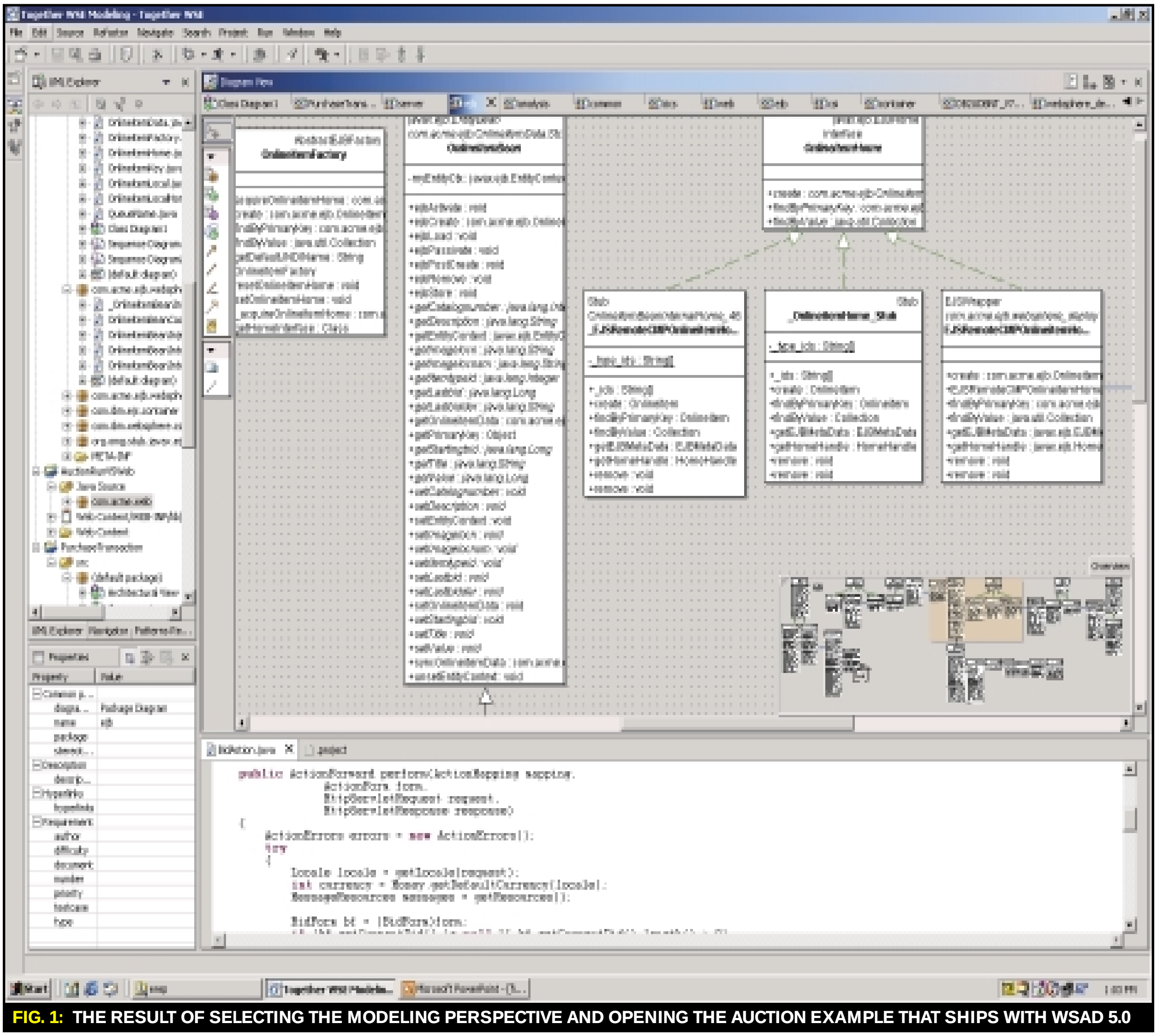
Concurrent Processing Using Message-Driven Beans

BY AMIT PODDAR

CASE STUDY:

Discovering and Documenting Business Application Patterns

BY BRENT CARLSON AND JAMES CAREY



```
public ActionForward perform(ActionMapping mapping,
                ActionForm form,
                HttpServletRequest request,
                HttpServletResponse response)
{
    ActionErrors errors = new ActionErrors();
    try
    {
        Locale locale = getLocale(request);
        int currency = Money.getDefaultCurrency(locale);
        MessageResources messages = getResources();

        BidForm bf = (BidForm)form;
```

FIG. 1: THE RESULT OF SELECTING THE MODELING PERSPECTIVE AND OPENING THE AUCTION EXAMPLE THAT SHIPS WITH WSAD 5.0

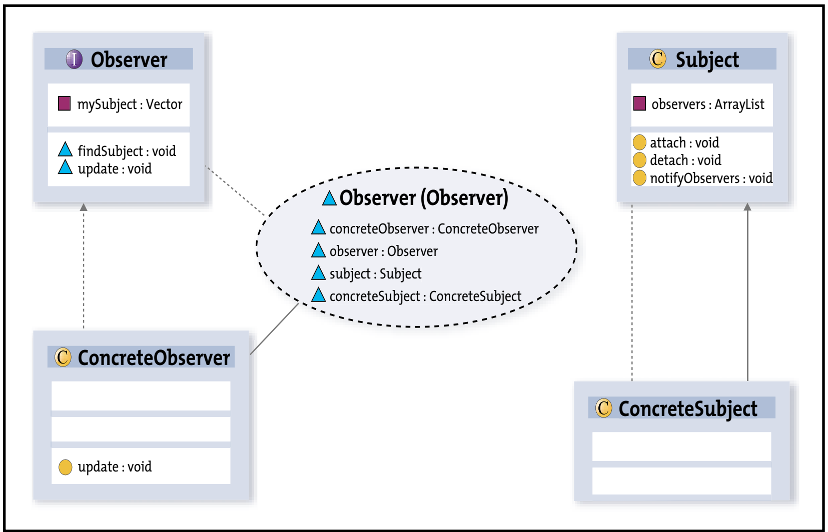


FIG. 2: TEMPLATES ALLOW NEW PATTERNS TO BE CREATED AS USERS SEE THE NEED FOR THEM

Profiles

UML Profiles allow extensions to the UML to be turned on and off. Currently, Together Edition provides UML Modeling in Color, UML Profile for Business Modeling, UML 2.0, Erikson-Penker Business Extension Profile, and a profile for Software Development Processes. In the future, you will able to add your own editions.

Together Edition is geared toward the IBM WebSphere environment. It is built on the Eclipse technology that powers IBM's WSAD. Together Edition for WebSphere Studio and Together ControlCenter are part of the same product family. I have used both, and they have a somewhat different look and feel, in order to appeal to both current users and WSAD developers.

A LIMITED TIME SAVINGS OFFER FROM SYS-CON Media

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!

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

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

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

□ **Linux Business & Technology**

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

□ **Java Developer's Journal**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Web Services Journal**

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

□ **.NET Developer's Journal**

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

□ **XML-Journal**

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

□ **WebLogic Developer's Journal**

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

□ **ColdFusion Developer's Journal**

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

□ **Wireless Business & Technology**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **WebSphere Developer's Journal**

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

□ **PowerBuilder Developer's Journal**

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

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY
WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $275^00^
Pay only $175 for a 1 year subscription plus a **FREE CD**

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350^00^
Pay only $395 for a 1 year subscription plus **2 FREE CDs**

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

Pick 9 of our magazines and save up to $400^00^
Pay only $495 for a 1 year subscription plus **3 FREE CDs**

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

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

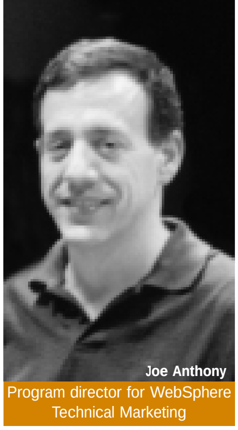
SYS-CON MEDIA

Scott Hebner
Program director for WebSphere infrastructure software

Stefan Van Overtveldt
Program director for WebSphere Technical Marketing

Aimee Munsell
Program director for WebSphere Application Server

Bernie Spang
Program director for WebSphere Studio Marketing


Joe Anthony
Program director for WebSphere Technical Marketing

Derek Bildfell
Program director for business development

The Evolving Role of WebSphere

A conversation with the WebSphere marketing team

Development tools like WebSphere Studio, according to IBM's Scott Hebner, have changed "from being almost rogue initiatives with a lot of customers driven by a couple of departments into really being strategic investments now." And the entire usage of tools like this, and of application servers overall, has, he notes, changed.

WebSphere Developer's Journal editor-in-chief Jack Martin had the opportunity recently to engage with Hebner's entire team, the team behind marketing IBM's WebSphere infrastructure software, in a wide-ranging and exclusive discussion. In this first installment, topics raised include networking challenges, Web services, RAD, the implications of the Eclipse technology pioneered by IBM two years ago, and the dynamics of IBM's on-demand application infrastructure. Most surprising of all is the revelation that WebSphere Studio will soon support the development, testing, and deployment of WebLogic applications.

WSDJ: What I'd like to do is to go around the room and have each of you tell me a little about your job.

Van Overtveldt: Basically, I am sort of the external antenna for the WebSphere marketing team. I work a lot with customers, analysts, and press, first to give them a better understanding of our products and our strategy, but also to learn from them, analyze their requirements, and feed that back into the product teams.

Munsell: I manage the product strategy and marketing for the application server, so I'm really a translator both ways between the development team and the customer. My job is to take what we've created and help customers understand how it helps solve their problems, as well as drive the customer needs back into the development process.

Anthony: I focus on our strategic initiative areas like the edge of network, autonomics, and grid computing, and identify how we are going to take WebSphere and get it out into the extended environment, particularly in the networking area.

Spang: I lead the team that is focused on product marketing for the WebSphere Studio portfolio. Like Amy, I play the role between development and the customer, taking the story out and bringing the requirements back, leading the strategy that's transitioned our distinct tools into a single portfolio with a strategy and a goal for the organization.

Bildfell: I spend a good amount of my time out doing strategic development deals with some of our key business partners that include Macromedia, Rational, Prolifics, and a series of others.

Hebner: I help to lead the portfolio strategies for the WebSphere foundation and tools. That is, the WebSphere Application Server and Studio development environments. Essentially, we drive product requirements, positioning, go-to-market, revenue growth plans, and overall business priorities for these products.

WSDJ: You're coming off a terrific accomplishment of blowing past BEA and Borland over the past couple of years. What do you think are the three biggest changes that have happened in the space over the past couple of years since you have been involved.

Van Overtveldt: The biggest thing for me is that the entire usage of application servers and development tools like WebSphere Studio has changed from being almost rogue initiatives with a lot of customers driven by a couple of departments into really being strategic investments now. Companies are betting their business on this type of technology. They want it to perform; they want it to be manageable; and they want it to run 24/7. That drives a different approach to how you go to market, how you build your products, and how you test your products before you launch them. That, combined with the effort we put in place on getting products out that meet those changed criteria, is a big reason for the success of WebSphere Application Server.

WSDJ: Bernie, from the tools perspective, what has happened over the past couple of years that's significant?

Spang: The most significant thing is the emergence of an open development platform similar to the open server platform with the Web standards and J2EE that's been in place for a number of years. The Eclipse technology we helped launch in 2001 established for the first time an open platform for development tools that all the vendors can build on and extend in the same way that application providers do on the server side.

WSDJ: Was this team the actual muscle that pushed Eclipse?

Spang: This is the team that pushed Eclipse out the door, established the messages and the strategy, and helped launch the consortium itself. We have to give credit to the significant development teams in Ottawa, Raleigh, and Toronto that conceived of, designed, and implemented the technology.

WSDJ: But you're the people who literally got the message into developers' heads and ISVs that Eclipse was a viable way of going about doing things.

Spang: Yes. And set the strategy to do this as an open project, an open consortium in which IBM would have one vote of many – we decided that this was the right way to do it, as opposed to making a proprietary development platform play like some of our competitors have.

WSDJ: Joe, you work with strategies. Were you part of that Eclipse play?

Anthony: Eclipse was driven by Bernie and his team. My role was getting the marketing team focused on questions like, "Which open standards should we go after?" "What are the common things like Web services that are key if you want to accelerate market adoption and customer acceptance?" We didn't want customers fearful of adopting it. If it has some good buzz, if the marketplace is rallying around it, and you have a good partner ecosystem, it will drive WebSphere demand. A focused market results in a much bigger pie for all the competitors to go after. We feel we are delivering a lot more value to the marketplace as a result of the entire ecosystem that we put in place and that will feed our overall growth. So, it's good for us, but it's much better for the overall marketplace as well.

WSDJ: Let me ask you a question that is unique to someone like you as a strategist in this game. When you first heard of the Eclipse concept, how did you view it?

Anthony: It was a great way to get the industry focused. The problem with all this new technology is that while everyone is off playing their own game, the customers get very confused, and they hold off on implementation. This was a great way to get everyone rallied around a common set of objectives with each of the partners bringing their key value-add and delivering the best final total solution to the customer, which is a win-win for everyone involved.

WSDJ: So, you saw it from an economic standpoint first, then a technological standpoint?

Anthony: Right. Because ultimately it is a business. Technology is nice, but if you focus only on technology, it's a going-out-of-business strategy.

WSDJ: Aimee, in the past couple of years, what have you seen come and go?

Munsell: I think the biggest evolution in the application server space is the transition from a stand-alone point product to a broader infrastructure platform. A lot of customers had a lot of individual projects that they had to deploy very quickly and then they realized that they had to connect all these projects or applications together more easily and reliably; they needed a consistent infrastructure. The impact has been that we have had to figure out how to address a very broad set of market and product requirements. We have to focus on how to give a wide range of different applications – whether they are ERP or CRM or whatever– a common operating system for the Internet. Being able to provide that kind of value to partners, as well as bring together the different products across IBM's software portfolio and give them a common foundation for better-integrated, yet flexible, deployment is good for our customers and good for IBM.

WSDJ: Are you involved with embedding Web services into Application Server?

Munsell: Yes. I worked on the initial Web services standards launch, which involved working with Microsoft and agreeing on common standards and introducing those standards into the marketplace. Simultaneously, we worked hard to drive adoption of the standards and value-add productivity tools for Web services into IBM products, particularly WebSphere.

WSDJ: So you had to go toe-to-toe with Microsoft getting them to agree with something?

Munsell: Yes, exactly. It was a challenging project, but in order to continue to build on the open standards that were going to provide the full level of functionality needed to do transactions across different applications and in the business-to-business arena, we thought that was a critical move, so we worked with Microsoft to bring those standards to market, and of course, to drive productivity tools and features to make a services-oriented architecture really accessible and useful for customers

WSDJ: In the industry right now, Web services is probably the single most important thing happening – everyone is busy working on something right now. From your point of view – and Microsoft is notorious for disagreeing just for the sake of disagreeing – why do you think you got them to agree?

Munsell: I don't think that they really had that much of a choice, because no matter what they would like, customers have multiple operating systems and platforms, and the picture gets even more complex when you look at what needs to happen in order to get integration across businesses to work. In order to show some progress at solving these types of customer problems, they had to make a step in that direction.

WSDJ: Anything new or remarkable due in 2003?

Spang: Can I give you an idea about what's coming? Everything will be focused on rapid application development and the productivity that comes with it. Particularly around a service-oriented architecture in which you are composing and choreographing services within a network without understanding or caring about the technology the provider has used to implement the service. Second, there are going to be hundreds and hundreds of specialized plug-ins from a variety of different partners out in the marketplace as well as from all the different IBM middleware and software products. They are going to add value and enable customization and personalization of the single, open development environment. Third, we are not going to support just WebSphere, but also platforms such as WebLogic with customized plug-ins.

WSDJ: Can you say that again? You're going to support WebLogic?

Spang: Right. WebSphere Studio will support development, testing, and deployment of WebLogic applications.

WSDJ: So WebLogic is going to have its own design tool set now?

Spang: It is. Well, not its "own" – we are sharing WebSphere's. We are all for the support, unification, and simplification of the customer IT environment so that they can go to a single development environment and deploy to multiple deployment environments. WebSphere Studio's integrated support for developing COBOL for CICS or RPG for iSeries applications are other good examples.

WSDJ: Does BEA know this yet?

Spang: They do now.

WSDJ: Okay. They know it. Because BEA tool strategies have failed; everyone knows that across the industry.

Munsell: It's because we listen to our customers. We have customers who have heterogeneous environments. They need development tools to work across multiple products. Java and Web services help with application portability, but everyone knows the vendors need to do some extra work to really make interoperability and migration – if that is what the customer wants – cost-effective. Another example is the proposed Rational acquisition. Rational supports .NET, and we are not going to discontinue that. In fact, the capability helps us meet our goal of being the best platform for integrating and running a mixed .NET and Java environment.

Spang: The bottom line is that we have customers who don't want to have to have different development environments for the WebLogic Servers they still have versus WebSphere.

WSDJ: That's going to raise a lot of eyebrows when people read this. Who is going to be the person who will talk about how your design tools work with WebLogic?

Spang: It will fall to me and Stefan – our teams and the extended organization.

Hebner: There are many customers out there that have WebLogic in the mix. The WebSphere Studio product is famous for its superior productivity for building to the WebSphere servers due to its tightly integrated development and deployment capabilities. Now, with similar support for WebLogic, Studio will probably become the most productive development environment for WebLogic out and will allow WebLogic customers to leverage hundreds of Eclipse plug-ins, which they cannot do today. Also, this is going to provide a single development environment for customers who have mixed app server environments and will provide WebLogic customers a solid migration tool to move away from WebLogic. We anticipate that a great many will do so.

WSDJ: Tell me about the Grid net initiative for Application Server.

Anthony: Grid is a very nice play. About a year ago, the grid infrastructure community adopted Web services as their methodology for intercommunication. As a result, WebSphere will be grid enabled, and the Globus Toolkit 3.0 that is coming out has a reference implementation based on WebSphere. IBM is fully supportive of what they are doing and is an active contributor to the grid community efforts. IBM recently announced the availability of a number of grid solutions that contain WebSphere Application Server, WebSphere Studio, and WebSphere Portal.

We are looking at how we support grid infrastructures; in moving forward, we will continue to add more features into the base products. Right now there are a lot of services that go into some of the total solutions that are provided. We will continue to take more of the grid functionality and build it right into the WebSphere infrastructure.

WSDJ: Do you have any business examples yet for Grid? Have you gotten that far yet?

Anthony: Certainly. In the petroleum industry, for example, they are doing seismic analysis for oil fields. They are taking a lot of raw data, and increasing the accuracy of their modeling efforts to determine which are the best fields to go after, which is critical in making oil and gas exploration investments. They are leveraging grid infrastructures to do that. Grid is also being used in the pharmaceutical and finance industries, life sciences, governments, and the industrial sector.

WSDJ: Are they using algorithmic information for that?

Anthony: There are a number of fields that are very data intensive or have applications that can be very highly parallelized, so the grid infrastructure can leverage that. We feel going forward that we can make the individual scientist a lot more productive. Today a lot of the grid applications are still pretty much hand-coded. We will be making WebSphere Studio enhancements that will make their lives more productive as they try to leverage these kinds of infrastructures.

Spang: There's an important aspect of the Web services initiative that explains why this team has come together and why we were able to accomplish what we were able to accomplish in the past few years.

Amy and I worked the Web services launch together as part of that team, and Stefan, Scott, and I go back together to the early days of Java and creating the "eBusiness" name and the whole Internet launch point. This team collectively has a lot of history around open standards and IBM's Internet play evolving into "eBusiness" – that's what we were able to bring to this team.

WSDJ: Scott, from your point of view, what are the three biggest things that have happened over the past couple of years?

Hebner: We have fundamentally transformed the Application Server and Studio development environment into a next-generation platform optimized for the dynamics of an on-demand application infrastructure. We became obsessive about understanding how customers use and purchase these products, and their ROI goals. This resulted in deeper insight to anticipate new requirements and usage occasions such as autonomics, Web services–oriented applications, business integration, server consolidation, and the requirements of creating approachable "on-ramps" to e-business. These insights drove the new features and configurations that are now available in v5, especially WAS Enterprise.

The second thing we were able to do was to greatly simplify the portfolio. From the product perspective, we took a variety of fragmented products and drove them into fundamentally two products that now have flexible configurations. We also simplified the naming, the pricing, and how we sell and partner with these products. I think the overall simplification of the portfolio helped to drive some of the success. Third, we were able to drive the value proposition within the context of a broader platform and the role that the application server and the tools played in the broader IBM infrastructure, including business integration, Portal, and commerce. Those were the three key things, I think, as I look back, that helped transport the products to where they are today, into a next-gen–optimized portfolio.

Next Month

In Part 2 of this interview the WebSphere marketing team discusses the importance of the customer in development and WebSphere's focus on delivering tools for rapid development.

"We are not going to support just WebSphere, but also platforms such as WebLogic with customized plug-ins"

Bernie Spang

from Macromedia

ColdFusion MX for IBM WebSphere Application Server

REVIEWED BY
ANTONIO VILLAFANA

The fusion of IBM's WebSphere Application Server (WAS) and Macromedia's ColdFusion MX (CFMX) brings together two proven technologies. CFMX for the WAS runtime environment is a Java application expected to take advantage of WAS's industry-leading J2EE platform implementation and superior technologies for processing and managing Web application requests, and managing database connections. CFMX for J2EE applications can now take advantage of the scalability, robustness, security, and portability of the J2EE platform.

ABOUT THE AUTHOR

Antonio Villafana, president of e-Mind Solutions, has over 12 years' experience in the computing industry. His wide-ranging experience includes 10 years of military service, IBM certifications, software design, and business process automation.

E-MAIL

villafana@attbi.com

WAS provides a highly scalable and well-managed environment for CFMX applications, performance monitoring tools for CFMX administrators, and application server–specific features such as application security, transaction management, database connection management, and access to message-oriented middleware systems. CFMX developers need only be concerned with the application server–supported J2EE platform level, and application server–specific implementations such as J2EE platform extensions and caching mechanisms. For developers, WAS provides the necessary interface for the deployment of CFMX enterprise applications. Applications can be deployed using a single enterprise application archive (EAR) or two Web application archive (WAR) files.

Fusion of Technologies

Although my initial attempt at the installation and configuration of CFMX for WAS 5.0 was a bit eventful, I was able to perform a complete installation and configure a test environment with relative ease. The complete installation is a three-step process, with an optional fourth step for enabling specific ColdFusion applications' functionality.

Before you can deploy any CFMX J2EE applications to WAS, you must first install the CFMX J2EE core enterprise application files. During installation I was prompted to choose whether or not my application deployment was going to be installed as a Web application resource bundle or enterprise application resource bundle (see Figure 1). Unless you are absolutely comfortable with your J2EE platform skills, I would suggest selecting the EAR option. Installing CFMX as a WAR file is much more complex, as it involves unpacking the EAR file and making some additional configuration changes in the WAS Admin console.

When prompted, I kept the default value (cfmx) as the name of my context root. The name assigned to the context root will be used, along with your domain name in the URL, by the Web server for accessing CFML (ColdFusion Markup Language) templates. Once installation is complete you can use http://www.mydomain.com/cfmx to test your installation and access your CFML templates.

During installation of the CFMX files a very detailed log of your installation is written to a file, Macromedia_ColdFusion_MX_for_J2EE_InstallLog.log, and the core CFMX J2EE EAR file, cfusion.ear, can be found in the CFMX root directory that you specified during installation. The log file is a great resource for locating default settings that you may not have kept track of during installation (yes…I had to use it).

The second phase of the installation involved deploying the CFMX J2EE files to WAS. Before continuing with this part of the installation you will be required to launch the WAS admin console (see Figure 2). To start the deployment process you must first navigate to Applications–>Enterprise Application, then click "Install." The "Preparing Install" page will launch, and you must enter the path to the <cfmx_ root>\cfusion.ear file. On my test machine this path was C:\CFMXJ2ee\cfusion.ear.

If you are new to WAS and/or deployment of J2EE enterprise applications, I recommend accepting the default values, when prompted, for the rest of the CFMX J2EE enterprise application installation process. If you have decided to enable Remote Development Services (RDS) support, ensure that RDS and CFMX are on the same virtual host and port. RDS gives developers using either Macromedia Dreamweaver MX or HomeSite Plus the ability to browse resources on the server running ColdFusion from within their IDE. However, you must consider the security implications of enabling RDS, and Macromedia does not recommend installing it in a production environment.

Once the deployment of a CFMX J2EE enterprise application is complete, the core CFMX J2EE installation files can be found in the installedApps folder of the node hosting your WAS. For example, on my test machine the core CFMX J2EE installed files are located in <was_root>\installedApps\ thinkpad\ Macromedia ColdFusion MX.ear.

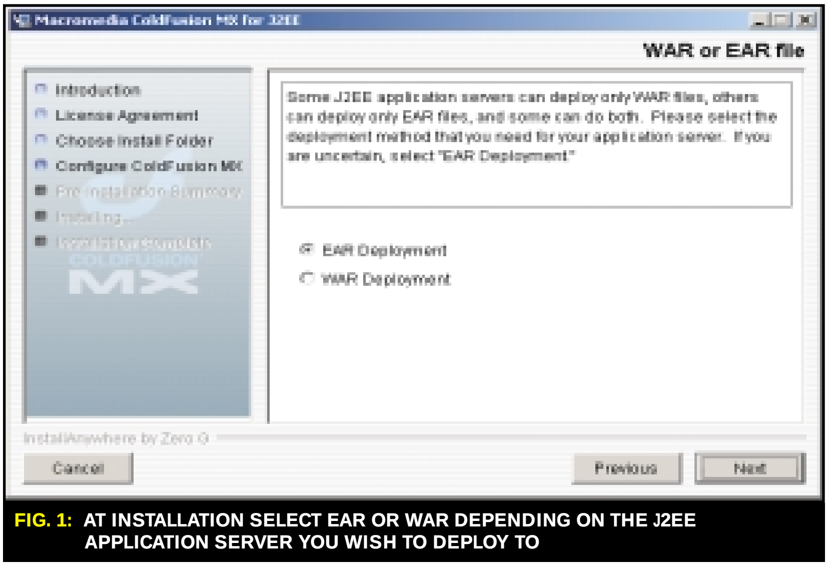
Once the CFMX J2EE engine has been installed, you are ready to run the startup wizard to configure CFMX. To run the startup wizard you will need to start the CFMX enterprise application. The newly installed Macromedia ColdFusion MX should now be visible in the WAS Admin Console Applications–>Enterprise Applications pane. Select “Macromedia ColdFusion MX” and click “Start.”.

Once the status display for the Macromedia ColdFusion MX application has changed to running, you are ready to launch the startup wizard. I launched the wizard by using the following URL: http://localhost:9080/cfmx/CFIDE/administrator/index.cfm.

For the final and optional step – before some CFMX-specific functionalities such as CORBA support, accessing COM objects, and using third-party JDBC drivers are available to applications, they must first be enabled, with each functionality having its own series of steps to perform before it can be fully enabled. The installed <cfmx_root>\Installing_ Using_CFMX_for_J2EE.pdf contains detailed instructions for enabling each functionality.

The CFMX Web-based admin console has all the basic application server tuning and configuration knobs, such as memory management, RDBMS connection management services, debugging, and security. The admin console for CFMX has resource locators for both developers and sys admins. This can lend itself to a somewhat cluttered view, especially for the more experienced system administrator, who is only concerned with configuration, debugging, and monitoring of resources. You may have to look through the admin console main page several times before finding the resource or configuration parameter you need, but once located, there is ample on-screen information for every configurable resource.

My initial mental questions on performance were “How long will CFMX applications take to load?” and “How will it degrade WAS performance?” After launching my first sample application, I started to wonder “How fast can this thing go?” I compared the response times of the CFMX sample applications versus deployed WAS J2EE sample applications, and decided to include just the fact that there was no notable difference in response times between the applications. This shines brightly on the abilities and careful fusion of these industry-leading technologies.



Macromedia ColdFusion MX for J2EE

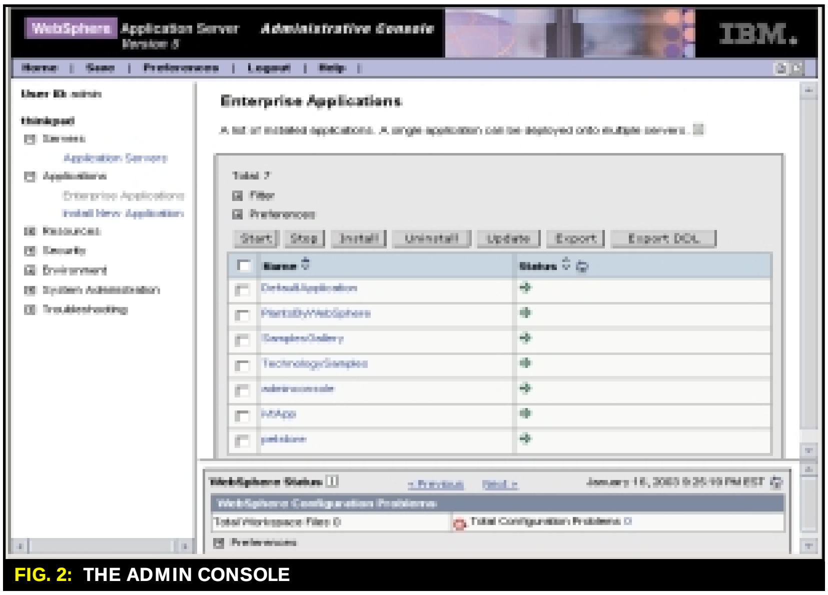
WAR or EAR file

Some J2EE application servers can deploy only WAR files, others can deploy only EAR files, and some can do both. Please select the deployment method that you need for your application server. If you are uncertain, select "EAR Deployment."

● EAR Deployment

○ WAR Deployment

FIG. 1: AT INSTALLATION SELECT EAR OR WAR DEPENDING ON THE J2EE APPLICATION SERVER YOU WISH TO DEPLOY TO



Enterprise Applications

A list of installed applications. A single application can be deployed onto multiple servers.

Total 7

⊞ Filter

⊞ Preferences

□	Name	Status
□	DefaultApplication	➜
□	PlantsByWebSphere	➜
□	SamplesGallery	➜
□	TechnologySamples	➜
□	adminconsole	➜
□	ivtApp	➜
□	petstore	➜

WebSphere Status < Previous Next > January 16, 2003 9:26:19 PM EST

WebSphere Configuration Problems

Total Workspace Files 0 Total Configuration Problems 0

⊞ Preferences

FIG. 2: THE ADMIN CONSOLE

Conclusion

For ColdFusion developers, CFMX for J2EE has opened up a whole new world of coding bliss and functionality, and for organizations using WebSphere, CFMX provides an easy-to-use rapid Web application development environment that takes advantage of the powerful WebSphere engine. The administration of the CFMX component and the performance of CFMX J2EE applications could be the most telling factors behind why current ColdFusion developers will make the move to developing full-featured J2EE apps, enabling them to take advantage of the J2EE platform and WAS. By choosing not to reinvent the wheel and add another application server to the market, Macromedia has appealingly kept the learning curve at a minimum for CFMX developers and system administrators.

Product Info:

COMPANY INFO:
Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94013
E-mail:
direct@macromedia.com
Web:
www.macromedia.com
Telephone: 800-457-1774

SYSTEM REQUIREMENTS:
WebSphere Application Server version 5.0

PRICING:
ColdFusion MX for WebSphere, 1 CPU
$3,399
ColdFusion MX for WebSphere, 2 CPUs
$6,798
ColdFusion MX for WebSphere, 1 CPU with 2-year subscription
$4,759
ColdFusion MX for WebSphere, 2 CPUs with 2-year subscription
$9,518

Managing Digital Identities

BY JOHN WORRALL

Organizations have traditionally viewed the management of digital identities as a necessary expense. But now those same organizations see it as a strategic imperative, something that will help them succeed by adding business value. Strategic security initiatives, like identity and access management, present organizations with real business opportunities and provide a framework for managing digital identities.

In today's competitive economy, security investments must contribute much more value to the organization's overall business by enabling key applications. Security is no longer just about mitigating risks by keeping the "bad guys" out. It's also about achieving key business objectives such as cost reduction and revenue generation, which can only happen by letting the "good guys" in. To this end, identity and access management delivers value and contributes to the overall success of the business.

The Definition

Identity management means different things to different people. Some define it as user data provisioning, while others think of it as user data storage. Both are critical components of any identity management system. But managing digital identities is also about intelligently using identities to achieve a business goal – whether that is increasing revenue, improving customer satisfaction, or reducing costs. And when conducting business online, organizations can only use trusted identities.

An effective identity management solution establishes trust in the organization's online environment and enforces its policies. Who users are (authentication) and what users can do (access management) are tightly coupled and absolutely critical. It's really about being able to manage the full life cycle of a digital identity from creation and maintenance to termination, as well as enforcement of organizational access policies. That's why the industry has redefined the term, using the more appropriate classification of *identity and access management*.

The Challenges

Organizations continue to invest in their e-business infrastructure as the primary platform for conducting their business. Key elements of this expansion include increased access to many resources using a proliferation of mobile devices. In addition, the growth of e-business will be characterized by redefined security perimeters with protection at the application level (not just the network) and adoption of Web services.

The expanding e-business model presents organizations with many daunting challenges – from a user as well as a management perspective. Some areas of particular concern include a cumbersome user experience, multisystem environments, multiple authentication requirements, complex authorization requirements, a shifting security perimeter, and increased exposure to risk.

The Solution

As an increasing number of applications are exposed to more and more users, organizations need to consider identity and access management requirements as they apply to their unique business objectives. The current approach of having separate authentication and access management infrastructures will probably not scale effectively as organizations automate more and more critical business processes. That's why organizations look toward the future, when all their needs can be addressed from one powerful system – the identity and access management platform.

Such a platform will provide these benefits:

- ***Improved user experience:*** An integrated solution will help users control their credentials, as they will no longer be required to manage a hoard of passwords. Nor will they be forced to sign in multiple times within a single session.
- ***Centralized administration and control:*** The right identity and access management solution will enable organizations to simplify the management of digital identities and security policies with one console – resulting in lower administrative costs and reduced resource burden.
- ***Multipurpose platform:*** An identity and access management solution will allow consolidation of solutions for multiple business challenges. Organizations will be able to manage both authentication and access management from a single platform.
- ***Enhanced integration:*** Seamless integration into an organization's heterogeneous e-business environment is critical. Identity and access management solutions will act much like middleware, enabling organizations to manage digital identities across their diverse and expanding infrastructure.
- ***Improved security:*** Identity and access management solutions will ensure greater levels of security to match the growing risk of exposure and high stakes involved in e-business infrastructures.

Conclusion

Intelligently managing digital identities is a vital way to gain a competitive edge and survive in challenging times. Identity and access management helps organizations take a holistic approach to security in order to have a direct, quantifiable effect on the bottom line.

ABOUT THE AUTHOR... John Worrall is vice president of worldwide marketing for RSA Security Inc.

E-MAIL... jworrall@rsasecurity.com