SYS-CON MEDIA

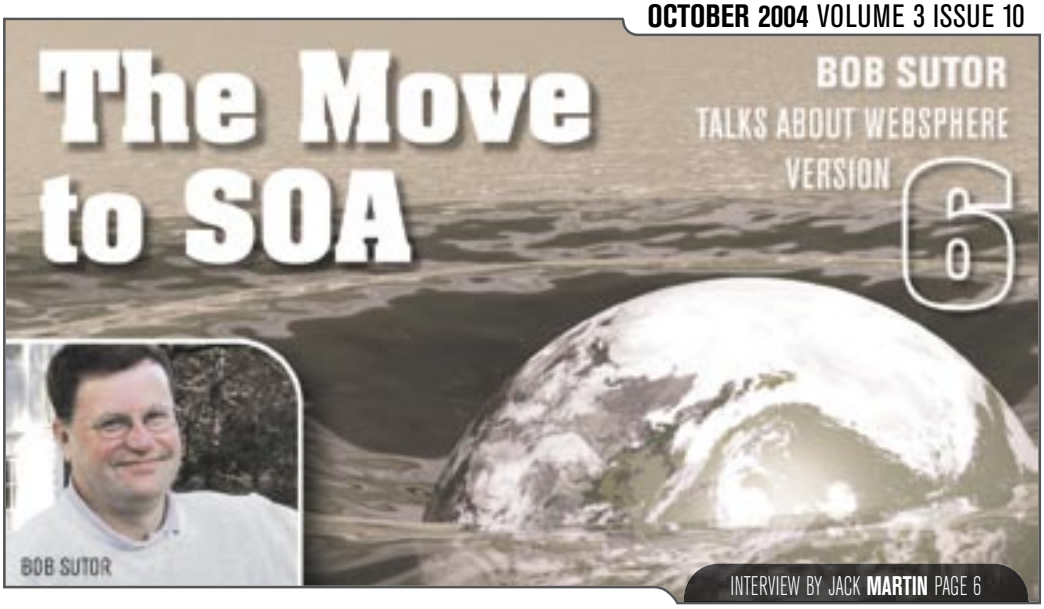
WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

OCTOBER 2004 VOLUME 3 ISSUE 10

DISPLAY UNTIL DECEMBER 31, 2004
$8.99US $9.99CAN
10
0 09281 03422 3



WebSphere Goes Autonomic

BY JACK MARTIN

IBM just announced WebSphere Application Server (WAS) version 6, which IBM says can save companies as much as $110,000 per minute in lost revenue and productivity. This capability is a major advance for enterprise-class computing,

WebSphere version 6 is designed to automatically detect problems – and automatically save and process Web-based business transactions that until recently could take hours or days to recover under older systems. It is also the first J2EE application server to deliver these capabilities together.

The financial impact of IT system downtime per hour varies by industry, but losses can quickly exceed millions of dollars per hour. According to IBM, independent studies cite the cost averages of $6.5 million per hour in the retail brokerage industry, $2.6 million in credit card authorizations, $90,000 in airline reservation centers, $27,000 in manufacturing, and $17,000 in banking.

IBM has put WAS v6 at the forefront of its autonomic strategy. This release enables a self-managed IT infrastructure with hardware and software that can configure, heal, optimize and protect itself. By taking care of many of the increasingly complex management requirements of enterprise-class systems, autonomic computing allows companies to focus their resources on business matters.

After detecting an outage, IBM WAS v6 automatically redirects data to a different designated "failover" server. That server can be within the same data center, or, in a more serious power failure or disaster, WAS can move the information via the Internet to a completely different location.

WebSphere developers will be delighted with an array of new wizards and a drag-and-drop environment that automates the most common and tedious steps of application development and deployment. By eliminating hand coding, developers can significantly reduce the number of programming steps previously needed to build an application.

Better scaling so more concurrent users can access an application will reduce administrative and licensing costs and provide better flexibility.

Web services standards are more automated and offer true cross-platform computing. WS-Security is supported, as are WS-Transactions and WS-I Basic Profile 1.1. Needless to say, this standards-based approach can respond quickly and effectively to business changes as part of a service-oriented architecture (SOA) strategy.

Leveraging SOA gives you the ability to interconnect business functions, processes, and services that can be mixed and matched on the fly using reusable, industry-standard software components rather manually-coded ones.

A key architectural component of an SOA is an enterprise service bus (ESB), which provides the connection infrastructure for business transactions to flow from application to application. WAS v6 dramatically simplifies the task of connecting WebSphere applications to an ESB. It features new messaging capability that performs faster than previous versions and integrates seamlessly with the existing IT infrastructure.

There is now greater consistency in the WAS family from top to bottom, including in the Express version. Features include support for J2EE 1.4 across the family, which makes it easier to develop and deploy applications using industry-standard tools.

IBM also announced that WebSphere Extended Deployment Version 5.1 – designed to automatically optimize the performance of companies' software and hardware, on demand, particularly during unexpected spikes in usage or changing market conditions – should be generally available on Oct. 22, 2004.

New versions of WebSphere Studio Site Developer and WebSphere Studio Application Developer are in the pipeline, rebranded as IBM Rational Web Developer for WebSphere Software and Rational Application Developer for WebSphere Software.

Jack Martin, editor-in-chief of ***WebSphere 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. Jack is coauthor of ***Understanding WebSphere***, from Prentice Hall. jack@sys-con.com

WebSphere® JOURNAL

ADVISORY BOARD
Richard Arone, David Caddis, Mike Curwen, Devi Gupta, Lloyd Hagemo, Barbara Johnson, Shannon Lynd, James Martin, Doug Stephen, Victor Valle

EDITOR-IN-CHIEF JACK MARTIN
EXECUTIVE EDITOR GAIL SCHULTZ
CONTRIBUTING EDITOR PATTI MARTIN
PRODUCT REVIEW EDITOR JAY JOHNSON
EDITOR NANCY VALENTINE
ASSOCIATE EDITOR JAMIE MATUSOW
ASSISTANT EDITOR NATALIE CHARTERS

WRITERS IN THIS ISSUE
Roland Barcia, Guy Barlow, Christophe Begue, Praveen Chhangani, Petter Graff, Jay Johnson, Jack Martin, Tony Nadalin, Andy Roberts, Asim Saddal, Lauren States, Bob Sutor

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 KIRCAALI
PRESIDENT, SYS-CON EVENTS GRISHA DAVIDA
GROUP PUBLISHER JEREMY GEELAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, INFORMATION SYSTEMS ROBERT DIAMOND
VP, SALES & MARKETING MILES SILVERMAN
PRODUCTION CONSULTANT JIM MORGAN
FINANCIAL ANALYST JOAN LAROSE
ACCOUNTS RECEIVABLE SHANNON RYSZMA
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
ADVERTISING MARKETING DIRECTOR DENNIS LEAVEY
SALES ADVERTSING MANAGER MEGAN MUSSA
ADVERTISING SALES MANAGERS KRISTIN KUHNLE, DOROTHY GIL
CONFERENCE MANAGER LIN GOETZ
LEAD DESIGNER LOUIS F. CUFFARI
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS RICHARD SILVERBERG, TAMI BEATTY, ANDREA BADER
WEB DESIGNERS STEPHEN KILMURRAY, MATTHEW POLLOTTA
ONLINE EDITOR MARTIN WEZDECKI
CIRCULATION SERVICE COORDINATORS EDNA EARLE RUSSELL, LINDA LIPTON, MONIGUE FLOYD
CUSTOMER RELATIONS MANAGER BRUNI STAROPOLI

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® Journal (ISSN# 1535-6914) is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

NEWSSTAND DISTRIBUTION CONSULTANT:
GREGORY ASSOCIATES, W.R.D.S.
732 607-9941, BJGASSOCIATES@CS.COM

FOR LIST RENTAL INFORMATION:
KEVIN COLLOPY: 845 731-2684, KEVIN.COLLOPY@EDITHROMAN.COM
FRANK CIPOLLA: 845 731-3832, FRANK.CIPOLLA@EPOSTDIRECT.COM

Why is it that some Java guys are more relaxed than others?

These days, with everything from customer service to sales and distribution running on Java, keeping your enterprise applications available can be pretty stressful.

Unless of course, you've discovered the power of Wily. No other software offers our level of insight. Which means you'll be able to monitor transactions and collaborate with colleagues in real-time. Even more important, you'll be able to optimize performance across the entire application—end-to-end.

So put Wily to work. And keep performance problems from pushing you over the edge.

Get Wily.™

wily technology

1 888 GET WILY | wilytech.com

The Move to SOA

IBM releases WebSphere version 6

INTERVIEW BY JACK **MARTIN**

Jack Martin, editor-in-chief of ***WebSphere 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. Jack is coauthor of ***Understanding WebSphere***, from Prentice Hall. jack@sys-con.com

On October 6, 2004, IBM announced the latest release of WebSphere, version 6. The next day, Jack Martin, editor-in-chief of *WebSphere Journal*, sat down to talk with Dr. Bob Sutor, the director of WebSphere Foundation Software, about some of the new features in this release.

Websphere Journal: Bob, tell us about the latest news on WebSphere:
Bob Sutor: We're launching WebSphere Application Server version 6, the first major release in nearly two years, and it contains, really, a tremendous number of new features that we think our customers will really be excited about.

WJ: What's your favorite feature?
Bob: The new high-availability manager. You know, if the customer's really critical business applications are running on an application server, it can cost a tremendous amount of money if that server somehow becomes unavailable.

For example, there are really two areas that you're concerned about when servers become unavailable. First, if they're not there you can't give them new work. You need to very quickly shift that work elsewhere. The second area of concern is what do you do with those jobs that were going on when the server suddenly became unavailable?

Say you're doing a big international money exchange; is that caught in limbo? How long does it take you to get that up and running again? We've been able to reduce the downtime for some transactions from what might be 5 or 6 minutes to perhaps 10 seconds or less.

WJ: How are you doing that?
Bob: It's a number of things. It's new algorithms; it's the configuration of the WebSphere clusters; and it's making sure that they are configured with network addressable storage. It's having this new manager actually being part of WebSphere. In many cases, this gets rid of the need for external high-availability managers. Since it is right in WebSphere, it has a more innate understanding of what the app server is doing and so can manage and monitor many of the internal mechanisms.

WJ: Can you describe what this manager is and how it works?
Bob: First of all, it works with a number of servers. This is definitely a feature of the WebSphere Application Server Network Deployment product. This is an extension of the failover and workload management features we had before. What the manager is doing initially is monitoring the health of the different servers, noting if they become unresponsive in any of several different ways. When it determines that one or more servers – it could be an entire WebSphere cluster – is no longer available, it can shift the work that was intended for those servers somewhere else in that data center or perhaps to another data center far away. Then what it does is it goes in and it grabs the transaction log, assuming it is available in shared storage, and hands it off for completion of in-flight work. So the new features are about greater flexibility that yield better availability, and these translate to real business benefits.

WJ: Correct me if I'm wrong; it sounds like there can be significant cost savings involved with something like this.
Bob: That's right. The question I would ask of a given customer in a given industry is "If a business-critical server becomes completely unavailable, how much is it going to cost you?" This can vary quite a bit on the top end – just think financial services, retail brokerages – it can cost about $6.5 million an hour. At the low end, say consumer banking, it can cost $17,000 an hour. In that situation, those costs might be from unavailable online banking. A customer might pick up the phone and call a service center or branch. When you're working out the problem person-to-person that can get expensive pretty quickly. So anything we can do to reduce the downtime from 5 or 6 minutes to something that's just a few seconds can save our customers quite a bit of money.

WJ: I assume that this is all still J2EE, that no one needs to develop any new skills to use this?
Bob: Yes, first and foremost WebSphere v6 is a J2EE server, specifically J2EE 1.4.

WJ: When developers look at App Server 6, what are they going to see that's different?

Bob: As I just mentioned, the developer is going to see that it has the full support for J2EE 1.4. And that means, for example, support for things like JAX-RPC which connects Web services with Java artifacts. But what you're also going to see is a brand new Java Messaging Service [JMS] 1.1 engine. The new engine is completely written in Java now; in our previous release, in version 5, it was a combination of Java and some of our other messaging code. The new engine runs faster because it runs on the same Java Virtual Machine as the application and has been highly tuned.

WJ: So everything is running native now?

Bob: It's running native. And it's running closer to what uses it. So for those reasons in our tests we've seen performance improvement up to 5 times, which is significant.

The second thing they're going to see is that we've done a lot of work in improving the administrative console. In particular, you can configure the new Java messaging engine much more easily than you've been able to do in the past. One of the types of configurations that you can do now is to connect the JMS engine with a particular message queue. So let's say that you have a messaging backbone that's based on WebSphere MQ, and this is connecting to your different enterprise applications, your mainframes, and your databases. It's much easier now to tie in your application servers to this messaging backbone. It is very much, I would say, advancing the state of the art of what one can do around the notion of an enterprise service bus (ESB), which as many people know is very much a part of this evolving notion of a service-oriented architecture.

WJ: Right. It sounds to me like you may be leveraging the MQ experience into WebSphere?

Bob: Certainly. We now have over 10 years of experience with MQ. I mean, nobody knows how to do secure and reliable messaging better than IBM.

WJ: MQ is definitely the king of that space.

Bob: It's the king. And we are always leveraging this experience in any way we can. Likewise, there is evidence of that completely through the J2EE stack. What we try to do really is give people the best of both worlds. Many people think of JMS as Java-to-Java communications. For the messaging backbone or MQ types of things, people think of connecting COBOL applications with databases, that is, more generally enterprise application integration. So when you can connect Java communications to existing enterprise communications you get something very powerful. This kind of legacy enablement with new use of standards gives you something that is definitely greater than the sum of the parts.

WJ: Let me ask you a hypothetical: suppose I was an MQ shop but I was not a WebSphere App Server shop. Does App Server 6 have enough going on with messaging that I would give serious consideration to becoming a WebSphere shop?

Bob: First of all, I would give consideration to why you were not a WebSphere Application Server shop if you were using any app server at all.

WJ: Let's just say my brother worked for BEA and I gave him the order.

Bob: I would want to consider WebSphere certainly because of its new messaging engine and its connections with MQ, among many other reasons. I think that the WebSphere Application Server has the most sophisticated support for Java and Web services standards in the industry. And when you can connect that smoothly into the legacy infrastructure that you have, it's a very powerful statement.

WJ: IBM has taken a real forward leadership position on service-oriented architecture; has put the rubber to the road and actually delivered product. From what I see here, this looks like the first major product supporting SOA from IBM.

Bob: Actually, this is not the first. We have been leaders in Web services, a way of doing part of SOA, since the beginning, back in 2000. Just to give you an example: we announced with Microsoft the first Web services spec, SOAP 1.1, in April 2000, and within two days of that announcement we had code available for developers to download for free. In fact, it was on a Friday and over the weekend 400 developers downloaded that first Java-based SOAP engine. The next month we donated it to the Apache Software Foundation, thereby kick-starting the whole open source Web services work to follow. We put it in the next release of the WebSphere Application Server as well. So, in some cases, WebSphere products are on perhaps their third or fourth generation of Web services standards.

The product we introduced last April, WebSphere Business Integration, includes support for BPEL, Business Process Execution Language. That is opening the door for people doing what we call service choreography, essentially a form of service-oriented integration. That product had strong SOA features and demonstrated our commitment to delivering key SOA features for what we call the foundation of the platform. WebSphere is designed to build applications on top of it – some by us, some by customers and business partners. We estimate that there are approximately 3,000 ISV solutions that run on top of WebSphere. That's coming from about 1,300 partners that are certified with WebSphere skills, not to mention how many more there might be that just haven't gotten their certifications yet.

It's important that we include these capabilities in the application server because this is the base, this is the foundation, on which so many other people will build. They are expecting to leverage this plumbing that we are building into it. We're master plumbers!

WJ: When you look at Application Server 6, how do you see the migration going?

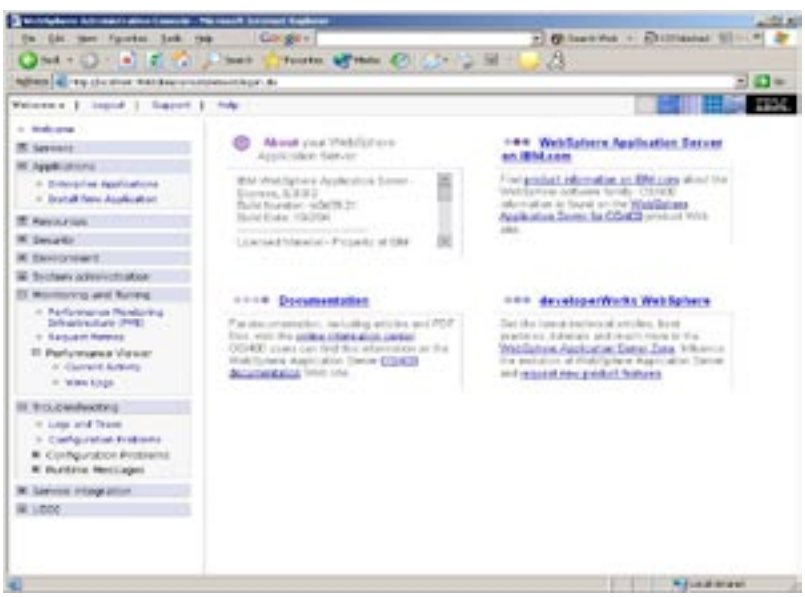
Bob: I think customers will migrate for several reasons. I think they will do so for the high-availability manager. They will migrate for J2EE 1.4. It's been a little while since J2EE 1.3 was new and I think a lot of customers are ready to move on, particularly because of the Web services support.

Version 6 has very good coexistence with version 5. By that I mean, let's say a customer is running a cluster of version 5 application servers. They're all in produc-

tion. Now, it can sometimes be a little scary for them to consider upgrading software if that means taking multiple servers out of production for a little while and then bringing them back up. It's time-consuming, and it's time that might mean there is some money lost. When you have a cluster of servers that are now at the version 5 level, you can systematically upgrade the servers one by one. Version 6 will happily understand that there are some version 5 servers and that the version 5 servers can do some things while the version 6 servers can do other things.

WJ: That's going to be very easy then.
Bob: I think for many customers it's going to be easier than its ever been in the past. Once they decide to make the jump because of the "must have" features in version 6, they'll be able to do it in a systematic way at their own rate.

WJ: You have a whole bunch of new wizards and all kinds of things to help people develop faster than they used to; that's a kind of standard thing that comes out of IBM every day to make developers' lives easier. But I get the sense that there is something very different going on.
Bob: We have really been spending a lot of time looking at the daily activities of developers who use WebSphere software. By that I mean that we are always asking ourselves what we can do to make developers' lives easier and how we can make those developers more efficient. One of the things that we've added in version 6 is what we call WebSphere Rapid Deployment. It basically says, you can go in, you can edit a Java file, and in order to deploy it, you can pick it up and drop it in a directory. We do ask that you add some hints to the source files so we understand what you intend that code to be for, say part of a certain type of Enterprise JavaBean. Once it goes into that directory, WebSphere can look at it and say, ah, this is a new file or it's an updated file; I know that I need to wrap it with a certain amount of code and compile it. I need to put it in a certain place and everything else it needs in the right locations. It really simplifies the process of writing and deploying code so that you can quickly test it and repeat the cycle if necessary.



There are other improvements in the tools themselves. We've improved the wizards. And in fact, we've improved the wizards so much that with many standard development activities, we've been able to improve the efficiency by about 75%.

WJ: Seventy-five percent is an enormous reduction. Where would someone be expected to find and pick that up right away?
Bob: You can expect to see that, for example, when you are starting a new project or a new piece of code within a project. Say you were going to create a Web service. Well, you can certainly start and you can write it all by hand with everything it might require. You might have a wizard that helps you do a little bit of this job. What we're saying is that we have gone through and we have systematically looked at the wizards that are in the tools. We've also tried to understand what additional wizards were necessary so that we can do a maximum amount of work for you, with your guidance. We hope in this way to make available the incredible features of WebSphere while ensuring that the simple things remain simple to do. You will always have to go in and customize your particular application, but we try to do the best job we can to get you as far along as possible.

WJ: And these wizards are located where? Do they come inside of App Server or inside of Studio?
Bob: They are in the tools. We rebranded some of the tools, by the way. WebSphere Studio Site Developer is now called Rational Web Developer for WebSphere Software. WebSphere Studio Application Developer is now called Rational Application Developer for WebSphere Software. The wizards live in those two tools.

WJ: When someone walks away from hearing this announcement, Bob, in your opinion, what are the three most important things they should know about WebSphere Application Server 6?
Bob: First of all, WebSphere Application Server is really part of the foundation for building an on demand application infrastructure.The features for high availability and secure, optimum use of resources are part of that.

The second is that version 6 has the key building blocks for service-oriented architecture. This product is going to be absolutely at the core of how our customers implement their service-oriented architecture. And that's by support of the latest Web services standards and optimizing use of the new Java messaging engine in addition to your current enterprise messaging solution.

Finally, I want people to walk away knowing that we're very focused on rapid development and deployment. We've tried to make sure that right out-of-the-box, this application server is tuned as much as practical in the ways our customers are going to want to use it. That is, we want to make sure that the default installation is going to be pretty close to what people want right from the very beginning.

Ignorance isn't bliss.

Knowing about performance problems before they happen is.

It might seem like bliss to ignore application performance problems and deal with them only if and when they surface. But waiting until a problem arises in production only results in crises, stress, and wasted time. Experience real bliss by eliminating performance problems proactively with DiagnoSys.

From development through production, DiagnoSys targets potential problems by providing a wide breadth of information, collecting data not only from the application server but also the associated resources, such as the Web server, database, and network.

This breadth of data gives a more accurate picture of a developing problem. Other products only look at the app server and imply performance of the resources based on the performance of the connections, a method that often misses the real source of the problem.

DiagnoSys also provides in-depth, statistically correlated information so you see a more complete view. DiagnoSys gathers information down to the method on the app server as well as in-depth information for all resources down to the OS.

Even though you get depth of information, you won't be overwhelmed. DiagnoSys synthesizes the data and quickly identifies the root cause so you can correct the problem before it impacts you or your company.

diagnoSys

Register for an upcoming Web seminar. Find ways to improve your Web app projects while driving profitability. Guest speakers share their knowledge. Register now at: www.hwcs.com/wsj16.asp

H&W

www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

Updating your wiring

The WebSphere/SOA Connection

BY BOB SUTOR

Dr. Bob Sutor is the director of WebSphere Foundation Software at IBM Corporation.
sutor@us.ibm.com

One of the major architectural themes for WebSphere Application Server Version 6 is its support for service-oriented architecture (SOA). IBM has supported Web services in various products for over four years, such as earlier versions of the application server as well as the WebSphere Business Integration Server Foundation, announced in April 2004. In v6, we have focused on support for standards such as J2EE 1.4, SOAP, WSDL, WS-Security, and WS-Transactions, and we've also made sure we had powerful, easy-to-use and easy-to-configure components such as the new JMS engine.

Though they are certainly connected, SOA is bigger than Web services. It is very reasonable to ask why there is all this interest in SOAs now versus, say, two years ago or two years hence. There are, in fact, several reasons, but let me focus on one in particular.

Web services appealed to a lot of people because it felt like a reasonable part 2 to the exploitation of the World Wide Web for communications between businesses and organizations. The Web set tremendous expectations in that when you put a URL in your browser, you didn't have to think about whether the page was going to be served up by WebSphere on Linux or z/OS, or BEA on Solaris, or Microsoft on Intel. Actually, that is not quite true because while I do 95% of my Web browsing using Firefox, there are still some sites that require Microsoft's Internet Explorer. Nevertheless, it is a great testament to standardization that so many sites work with so many browsers.

Before the Web came along, we had companies like Prodigy and AOL that had fairly closed environments. You called their phone numbers and you got their content. You had to use their clients to view that content and the navigation schemes were standard only to their worlds. The Web and standardized browsers changed all that. This is not to say that you can't get valuable additions by using AOL today, but you demand access to all the Web content as well, plus support for the standards.

So it was not unreasonable to ask for similar abstractions or loose coupling when we wanted to leverage the Internet and XML for program-to-program communication. If you are my business partner, I really don't care how you built your IT infrastructure as long as it is interoperable with mine and supports the quality of service we need to engage. That's why IBM helped create the Web Services Interoperability Organization and has helped drive the standards for SOAP, through WSDL, through security, transactions, reliable message delivery, and so on. There have been many, many companies and individuals who have contributed to the ongoing standardization of these standards. That has been essential to its success.

One of the things that IBM brought to the table was our experience in enterprise computing. Perhaps it would have been fun from a computer science perspective to take 20 years recreating all of the requirements and standards for interoperable and highly secure, reliable, and available services, but the reason why Web services has advanced as far as it has is because it has leveraged what we learned before. We've taken the good bits and tried to build those into a componentized system of standards that allow you to use only what you need, and we've tried to avoid the techniques and architectures that would lead to early failure.

So, I would argue a lot of the early motivation for building Web services was to take the enterprise experience and push it out to the Web. Of course we thought it would be useful inside companies as well, but we had technologies to handle most of the intra-enterprise requirements. Ultimately, we are after a set of standards and technologies that work seamlessly end-to-end from inside my organization, out through the firewall, maybe through intermediaries, through your firewall, and eventually to the endpoint application or service inside your world. To do this, we need to leverage the technology that is already installed and working at the quality of service levels we require. IT is all about legacy enablement, because the cool system you install today is tomorrow's legacy.

So here is part of my thesis: SOA is

hot now because it reflects a concerted effort to have a full end-to-end architecture that works within and between enterprises.

We are now using the early experiences and the maturing standards of Web services to work in the enterprise with traditional middleware products. To use IBM examples, that is why we have put so much effort into evolving and teaching products like CICS, WebSphere Application Server, WebSphere Business Integration Server Foundation, WebSphere MQ, and DB2 about Web services standards. These products are all highly useful for SOA use within and between organizations. We will continue to add support where it makes sense to our software product line.

SOA is of such great interest because we are now using the Web services technologies and standards to do a better job inside the enterprise, rather than the other way around. Two years ago we didn't have enough Web services standards for this to be viable. We can do SOA better now because there is enough in place to understand the principles and patterns of the overall architecture. Sure, we'll know more in two years, but we have enough to start tying all this together today. The programming models will continue to evolve to do a better job of simplifying how we access data and services, and the middleware will evolve to support those new models. Indeed, I would argue that programming models will continue to evolve to focus much more on SOA and less on language artifacts.

I mentioned earlier the new JMS engine in WebSphere Application Server v6. This new engine makes it easy to connect to a WebSphere MQ–based enterprise backbone. This means that we have evolved the application server to become an important part of what we might call the universal connection infrastructure, in addition to the usual enterprise messaging products that you might think of. Industry analysts and many vendors have settled on "enterprise service bus," or "ESB," as the name for this universal infrastructure.

I'm beginning to think we need another word after "enterprise service bus." The problem is that some people think that an ESB is a product and that causes confusion. Evidently, they believe you go and buy one and then you are all set. Your friend down the street also buys one and you both have the same thing. This is wrong. An ESB is not like a simple consumer item where two people can buy the same model and it fully meets all of their individual requirements.

An ESB is something you build for your enterprise or organization to give you the connection architecture you need to meet your IT and business goals. It can be built incrementally from multiple products and it needs to support the performance, reliability, and range of protocols that real, non-toy infrastructures require. If you have enterprise messaging products in place now or are about to install them, you have an ESB and a strong basis for future expansion and use of developing standards. To put it bluntly in IBM terms: if you use WebSphere MQ and other WebSphere brokers or integration servers, you have an ESB today.

Here's an analogy: our house was built in 1820, well before houses were wired for electricity. There wasn't much in the way of indoor plumbing in those days either, but that's another story. When it first became viable to wire houses, it was done with relatively primitive technology called "knob and tube." The positive and negative wires were separate, often running down different sides of the wood joists in the ceilings and the studs in the walls. Where you needed to affix a wire, you used a ceramic knob that was screwed or nailed to the wood. Where you needed to go through a joist or stud, you drilled a hole, put in a ceramic tube, and then ran a wire through that. Aside from not being grounded, the life of the wires was supposed to expire sometime in the 1950s. In our previous house I was replacing this old wiring with the new style well into the 1990s. I'm sure some of it is still there, but we moved, so it's not my problem any more.

Aside from the wires, the knobs, and the tubes, there were differences in how you wired a house then versus now. For one, you were lucky if each room had a single outlet (ungrounded, of course). The appliances were much simpler then, too, and I would wager that most of them were lights. Over time, people upgraded the wiring and added radios, refrigerators, microwaves, and computers. You can now even use household wiring to create a home network, something no one imagined when those first knobs and tubes were put in.

In theory, when new electrical codes come in (think new standards), everyone is supposed to rush right out and replace all the old stuff so the house doesn't burn down. In practice, this is done incrementally, to at least give you time to figure out if you really need to tear down that ceiling or wall to get the new wire through. In theory, you could just poke a hole and run a wire. Old houses are notorious for having obstructions that you wouldn't have guessed were there. In the meantime, the electrical system keeps working. Doing it incrementally allows you to budget for the work as well.

In the process of upgrading an electrical system you should probably add more outlets (most modern houses have outlets every six feet or so), and you may need to run a new electrical service to the house to get more amperage to power all the things plugged into all those shiny new outlets. The cost can quickly add up.

With the possible exception of brand new cookie-cutter development houses, condos, or apartments, residences have different configurations of electrical wiring. You don't go to a hardware store and buy a completely preconfigured set of wires that exactly fit the new power requirements and your house's architecture. You or your electrician (think consultant) buys a long length of wire and then cuts it to the proper length to connect to some appropriate point in the existing wiring infrastructure. As part of this, you probably installed a new junction box and outlet. Even then, there are

some choices to make because you may need to put in a GFCI outlet if the location is outside or near a wet area like a sink or shower. There are a few more variables, but I think you get the idea.

So although I've used a wiring example, there are a few concepts here that we can relate to IT:

- No one ever ends up with the original connectivity infrastructure that may have been planned on day one, because new requirements come in.
- Scalability is very important: you need an infrastructure that can handle the necessary throughput without starving critical business applications of the connectivity they need.
- The infrastructure needs to be incrementally extendable and compatible with what was previously installed.
- You need to be able to support a wide variety of qualities of service.
- Upward compatibility for applications is important.
- Standards are always being created and we need to be able to support both the new ones and the old ones, assuming they have not been supplanted for safety reasons. The implementation of the standards has to be accomplished in an evolutionary way to be practical.

Think of this home wiring example when you ponder ESBs. They are not a scary, futuristic topic. We and others in the industry have started to use the term in an umbrella way to talk about how we systematically simplify how we talk about enterprise connectivity that includes features for reliable message delivery, support for both new and legacy protocols and transports, data transformation, logging, high availability subject to connection requirements (that is, always connected, sometimes connected, rarely connected), and so on. We need this to support service, event, and traditional messaging infrastructure. It's easier to say "ESB" than everything else I mentioned in this paragraph.

Ideally, we could take some of the processing that originally had to take place at the endpoints and move it into the connection infrastructure (the "bus"). To use another electrical example, the IBM ThinkPad on which I am writing this article has a power supply that automatically switches between 110 and 220 volts. This means that, assuming I have the proper adapter, I can use the same laptop in the US or Europe without modifying the computer in any way. Users of early computers may remember a little switch on the power supply that you had to push one way or the other for either of the voltages. By the way, it's a really bad idea to plug a 110-volt device into a 220-volt electrical service.

Anyway, the little black box that sits between the electrical outlet in the wall and the plug I stick in the ThinkPad is akin to having intelligent functionality that is on the bus versus being at an endpoint. We've been able to move those smarts away from the endpoint, thereby simplifying it. In our IT example, if we can do data transformation on the bus, we don't have to teach all the applications or services that plug into the bus to do the transformation. This greatly simplifies the development of the applications. It also means that we can go to one or more designated places on the bus and add new transformations. Moreover, we can upgrade the physical servers on which the transformations take place in order to get better performance and thus scalability.

IBM customers who have been using WebSphere Business Integration Message Broker (and its otherwise branded predecessors) connected to applications via WebSphere MQ have been able to do this for years. These same products have been incrementally adding support for XML and Web services. We will continue to add features to our WebSphere products, as appropriate, to support the new standards. This means that the "bus" will support all the important things it does now, plus the new stuff as well.

So what do we call this concept to get people away from this single product notion? Here are some possibilities, and I'm only going to consider adding one or two words at the end of "Enterprise Service Bus," lest we confuse people even more:

- ESB Architecture (yuck; hard to talk about with SOA in the same breath)
- ESB Pattern
- ESB Infrastructure
- ESB Implementation
- ESB Network
- ESB Blueprint

Please let me know if any of these ring true to you or if you have better suggestions (I hope you do).

"Though they are certainly connected, SOA is bigger than Web services"

Conclusion

I've taken this discussion a long way from WebSphere Application Server v6, through SOA, and then into how to think about ESBs. WebSphere enables SOAs and provides many new and updated features to help you build a modern, standards-based, high performance, and highly available infrastructure that ties in and helps you better utilize your legacy investments.

Jack Martin, editor-in-chief of WebSphere Journal, recently sat down to chat with Rick Thomas about CICS

Why CICS?

RICK THOMAS
PROGRAM DIRECTOR FOR THE CICS TOOLS MARKETING

INTERVIEW BY JACK **MARTIN**

In this Q&A Rick discusses ways in which companies can protect their mission critical data even as they connect their networks with outside partners.

WebSphere Journal: Rick, what do you do over at IBM?
Rick Thomas: I'm the program director for the CICS Tools Marketing within the Application Integration Middleware division.

WJ: I understand that you have a pretty interesting new tool where you can get CICS and WebSphere to interoperate, which has been really hard for a lot of people. Tell me about it?
RT: This new tool is CICS Business Event Publisher. It's actually now in its second release with some interesting new capabilities. The Business Event Publisher is a tool that allows CICS to drive business processing in a mixed workload environment that includes CICS, IMS, WebSphere, WebSphere J2EE, WebSphere MQ and WBI Message Broker.

WJ: A lot of our readers are Java programmers, obviously, but not all of them are experienced with CICS. Who uses CICS today? How big a product is it? It's having its 35th anniversary.
RT: Yes, it is. We just celebrated our 35th anniversary of CICS. CICS is used by 10,000 customers worldwide and it is still the primary engine for business transactions everyday around the world. It's used in all the finance industries, insurance, retail, distribution for all purposes in all walks of life.

WJ: So that's something that's been sitting back there for a long time but a lot of people haven't migrated off it yet?
RT: That's right, and all indications are that they are not going to migrate. In fact, what they are going to do is revalidate the value of their CICS assets in mixed Web environments that include WebSphere with J2EE applications on one side and CICS applications on another.

WJ: What is the most important problem that this tool solves?
RT: The Business Event Publisher allows CICS to be in the driver's seat in business processes. Today, it is very common for customers to write WebSphere applications that drive CICS transactions.

They make a customer information request, operate on it in some way, and then submit an update of that customer information. That's an environment where CICS is acting as a partner with WebSphere, but the transaction is driven by WebSphere. The Business Event Publisher allows CICS essentially to be aware of actions that occur within CICS and know where to drive outbound business processes based on that. With the Business Event Publisher, CICS now gets to query WebSphere or a WebSphere database or another information database for customer information, get that information, operate on it, and send a response or an update outbound. Business Event Publisher actually allows CICS to have a more direct outbound business process–driving capability.

"It is still the primary engine for business transactions everyday around the world"

WJ: Very cool. I understand that you have a new tool to make it easier to migrate from VSAM, the database for CICS, over to DB2. Tell me about that?

RT: It's called the CICS VSAM Transparency Product. This is a product that we brought to market with our eye on supporting a critical mixed workload environment between WebSphere and CICS. Specifically, VSAM Transparency allows customers to migrate their mission-critical data from VSAM into a DB2 database.

WJ: That should be very helpful for a lot of people.

RT: It's pretty helpful for all those WebSphere J2EE application developers who are trying to access that same pool of legacy data.

WJ: That makes sense. There is a lot of that out there.

RT: What you're trying to do, if you are the customer, is enable the WebSphere application developer to access that data without causing the CICS programmer to rewrite all of his or her existing CICS applications. If you were to simply move all of your VSAM data into DB2, the customer would have to rewrite some or all of their CICS applications. The VSAM Transparency product allows CICS to run as if thinking that its data is still in VSAM and allows the WebSphere applications to run against the new DB2 database. Then everybody's happy and it allows customers to deploy interesting new WebSphere applications at a fraction of what would otherwise be a very expensive migration.

WJ: Finally, I'm not an expert on CICS, as everyone knows, but I understand that you have solved a significant problem with VSAM copying. Can you explain that?

RT: Yes. There are obviously CICS systems programmers who have been copying the VSAM data for as long as there has been CICS – 35 years. The problem is that most of the products in the marketplace today have required them to take CICS offline prior to doing these kinds of copies. We have come out with a tool called CICS VSAM Copy that operates uniquely with a copy while open for update function. The copy while open for update function means that CICS does not have to be taken offline to do the copy. Taking CICS offline affects WebSphere when you're using WebSphere as a front end to CICS. So in any environment where you have a WebSphere and CICS mixed workload application, customers will find this copy function to be extremely valuable. It means CICS will no longer have to be taken offline, which means the WebSphere application will no longer have to be taken offline.

WJ: Is there anything that you want to wrap up with?

RT: I'd just like to remind you and all your readers that this is still a growing environment. There are more CICS transactions every year than there were in previous years and it's been this way for 35 years. Likewise, we're now celebrating the 10th anniversary of MQ. So customers are continuing to see significant investment in both CICS and MQ environments.

On top of that, customers are now seeing the obvious value of their WebSphere Application Server as a platform for new J2EE applications. With CICS, WebSphere MQ, and WebSphere Application Server, you have a very powerful ecosystem in which to run business-critical applications. I think that the new CICS tools we are announcing now go a long way to further enable customers to take advantage of this significant CICS investment, and their investment in the new WebSphere Application Server.

WebSphere Journal Coming Next Month...

WIRELESS DEVICE MANAGEMENT
Necessary evil or opportunity?

IMPLEMENTING SARBANES-OXLEY
Using IBM Workplace for business controls and reporting

THE CHALLENGES FACING LIFE SCIENCES ORGANIZATIONS
IBM's middleware solutions do it all

THE SUPPORT OF COMPOSITE APPLICATIONS IN WEBSPHERE PORTAL
The basis for advanced suites of business functions

Delivering Applications on Demand in SOA Environments

Fast development with fewer resources

BY ANDY **ROBERTS**

Andy Roberts is vice president and chief technology officer for Bowstreet. Prior to joining Bowstreet, he was vice president of product design and planning for Parametric Technology Corporation, a highly successful provider of mechanical design automation solutions. One of the lead architects of PTC's product strategy, Andy designed and prototyped more than 20 shipped software products in the Pro/ENGINEER family, a product line that generates $1 billion in annual revenue. He holds a bachelor's degree in physics from Harvard University and a masters' degree in mechanical engineering from MIT.
aroberts@bowstreet.com

Application developers have always been under pressure to develop applications faster, with fewer resources. Recently, this has directed attention to service-oriented architectures (SOAs) because of their promise to provide libraries of reusable services that can be snapped together easily when needed.

While SOAs continue to evolve and gain adoption, application developers are also seeing limitations in just how far an SOA can go in delivering development speed at lower cost. SOAs are great at providing libraries of loosely coupled, reusable computing functions, but developers still need to code application infrastructures around these services in order to solve specific business scenarios. To address this problem, many infrastructure vendors have augmented SOAs with abstraction layers such as JavaServer Faces (JSF) that make it possible to write applications in fewer lines of code. Additionally, developers are increasingly sending development jobs to places like India, where labor is cheaper.

Each of these approaches to building and delivering applications faster and more cost effectively, however, is incremental in the long run. Ultimately, the goal is to deliver applications in real time, or on demand – just as fully automated, robot-empowered factories manufacture and deliver mass customized products like cars and electronic products in real time – without employing manual fabrication steps in the finished product. If you think of real-time, on-demand delivery of customized apps as the ultimate goal, then a new kind of software-based automated fabrication layer is required to augment SOAs.

SOA – A New Programming Model

The term SOA refers to the new kind of programming model that refocuses on the age-old problem of software reuse. Today's SOAs are an outgrowth of the Internet. They take advantage of XML and HTTP, and provide a new level of distribution and reuse of computing functionality that has never been seen before in software.

SOAs aren't intended to replace or "front-end" all existing back-end systems. They are intended only to replace the ones that require cross-platform access and standardized, self-describing interfaces (i.e., XML schemas). SOAs are currently evolving and are still in their early stages. As it stands, there are still many open issues with respect to how SOAs should, and can, provide basic functions around services such as security, failure, and contracts.

Today, because SOAs are so new, application developers are forced to use SOAs in conjunction with more traditional means of accessing application resources, such as tightly coupled Enterprise JavaBeans (EJBs), message queues, and language-specific, API-based connector architectures.

SOAs make it easier for business functions from different back-end systems to be seamlessly integrated into user-facing applications. This is because it is easier to transform data from one format to another when it is represented in XML, and accessed by way of independent, self-sufficient services. For example, an SOA can enable an SAP system to be linked with a Lotus Domino system at the application session layer to provide a comprehensive employee travel expense reporting system. Previously, two enterprise-level systems would have called for independent "silo" applications: SOAs enable a new breed of composite application to link services from disparate back ends into a common session running on an application server, such as J2EE-based WebSphere.

Application platforms are moving in a service-oriented direction as well. IBM, for example, is building its SOA strategy, in part, around WebSphere Application Server and integration software. The company has also established the WebSphere Business Integration Server Foundation, offering native support for the Business Process Execution Language (BPEL), as well as the IBM Assessments for Services-Oriented Architecture from IBM Global Services.

SOA – LEGO™ Blocks for Building Applications

SOAs enable developers to 1) reuse functional software components, and 2) construct "composite applications" out of these pieces as shown in Figure 1. Just as the LEGO™ block concept increased the space of what one could build with a relatively small set of off-the-shelf

pieces, SOAs enable developers to provide a wider array of applications out of a small set of pre-built services. The philosophy driving an SOA is that one can standardize and lock down a relatively small set of services that don't change that often, and still achieve a needed level of application variability by virtue of the large number of ways in which these parts can be assembled. Of course, there's still a certain amount of custom fabrication that's needed to build the small set of parts that are needed for the finished product or application. This is often referred to as the "last mile" of development, which can form a bottleneck that leads to its own problems. For now, SOAs provide the first step in simultaneously enabling the seemingly diametrically opposed requirements of simple standardization and reuse, versus mass customization.

SOA – A Prerequisite to Delivering Applications on Demand

The SOA model brings about the ability to assemble applications out of piece parts – i.e., services. The similarity of this model to the manufacturing world beckons people to embrace concepts such as "make to order" and "just in time." On demand signifies delivering an application when it is needed – on demand. Without an SOA, on demand is difficult, if not impossible due to the lack of off-the-shelf, reusable piece parts – i.e., services.

The term "on demand" used in the context above pertains to the delivery of applications when they are needed – on demand. There is another use of the term "on demand" as it pertains to the runtime operation of applications and networks. In this context, on demand means the ability to adapt automatically to changes while running. One can think of operational on demand as being analogous to a car that's able to adapt to changes in terrain by stiffening the suspension and changing the braking behavior in response to changing road conditions.

While on demand application operability is important, its on demand application delivery is what really takes advantage of the power of an SOA. However, in order to realize the full benefits of on demand application delivery and an SOA, another piece is needed – automation.

Factory Automation – Enabling Real Time On Demand

Even if a development shop has an SOA, delivering custom-built applications on demand still requires people to develop application code that makes use of an SOA-provided library of reusable back-end services. The application code that forms the "structure" of the application must be developed for each business need, and this requires development cycles for each scenario. In order to make delivery of custom-generated applications be truly on demand, an SOA platform needs an automated fabrication component linked to it – akin to a factory that uses robots to perform automated sequences of fabrication steps. Without some form of automated fabrication component to assist an SOA with the creation and delivery of finished applications, on demand runs into a bottleneck that's formed by the lack of human development resources needed to build finished applications. Even with the best of SOAs, stocked with libraries of off-the-shelf services, development shops must still pour large amounts of resources into building and maintaining applications that utilize these backend services.

Many development environments today are lured into trying to solve the human bottleneck problem by using a combination of RAD tools to assist developers in the construction of software components, and less expensive offshore services. Both efforts help in some respect, but fail to solve the larger problem – to deliver on demand in a real time manner. RAD tools enable the rapid creation of single instances of software components, but they don't perform the critical role of automating the creation of families of related, but different application instances when they are needed – on demand. In other words, RAD tools are not able to automate the generation of different instances of applications, in the same manner that a robot-enabled factory is able to manufacture families of

"SOAs enable developers to provide a wider array of applications out of a small set of pre-built services"

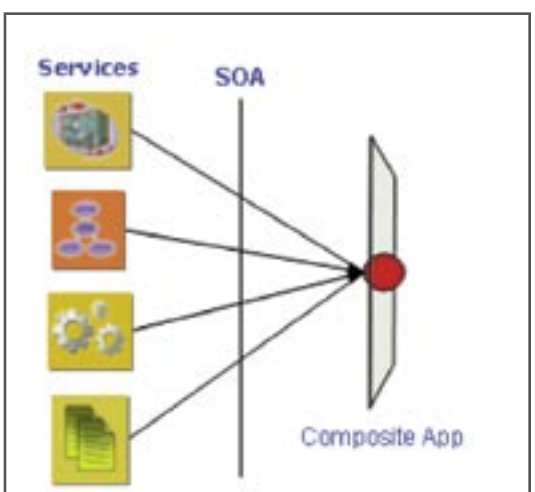


FIG 1: COMPOSITE APPLICATION

varying instances of products by varying the fabrication steps performed by individual robots. Instead, RAD tools are intended to assist and speed the manual creation of individual application instances. This makes both RAD tools and offshore services better suited for constructing the more static services that populate the warehouse library of an SOA.

In a physical world, an automated factory performs the task of fabricating the chassis and structures to hold "off-the-shelf" components that get assembled into a finished product. The analogy here to the services model is that services are analogous to off-the-shelf, reusable components, and the chassis is analogous to the application presentation, session logic, and workflow.

Enabling Many Variations from a Single Model

Today, the need for application variability and mass customization is growing stronger as developers are providing wider access to applications via the Web. Larger audiences translate to wider ranges of end-user requirements. As a result, the old Burger King slogan, "have it your way," is becoming a requirement of application services that are provided via the Web.

There are many different characteristics that can potentially cause an application to have its composition of feature and option combinations vary. These characteristics can be thought of as dimensions that drive an application's variability. Every application, even the simplest one, has at least one dimension of variability, and that is user role. Every application has users that fall into different usage roles, such as, a user role, an admin role, a developer role, and a tester role. Each of these roles causes the application to have to exhibit different functional behaviors.

In real business scenarios, the number of variability dimensions can grow into the dozens. For example, in a flight reservation booking application, there are several characteristics, such as the types of tasks that need to be performed, the level of entitlement for the end user, the end user's geographic territory, the look-and-feel requirements of the application, and the specific time in which the application is being used. Each of these dimensions creates the need for different functionality and structure to be woven into the application.

"Service providers are turning to a new application delivery solution called syndication"

Syndication – The Force that's Driving Variability Through the Roof

In certain industries, such as travel and financial services, application delivery via the Web is becoming increasingly complex. Previously, a service provider, such as an airline or hotel, could build a Web site and expect users to visit its site. Today, a sea of intermediaries, such as travel service packagers and aggregators, and special interest sites offer their own special interest packages. As a result, airlines and hotels are being disintermediated – causing a dilution of their brands on the Web – where consumers are making buying decisions based on price over brand.

In order to fight back and gain lost market share, service providers are turning to a new application delivery solution called *syndication*. To overcome challenges associated with trying to get users to visit an application exclusively at a single "home" site, a service provider can syndicate its application out to thousands of other sites. Syndication means that access to the application gets embedded into partner sites in a similar manner as to the way a banner ad is embedded in a site. The difference occurs when a syndicated application is accessed by an end user visiting the partner site and the syndicated application has to be served up on demand in a manner that is similar to how a banner ad is served up in the context of another site. This is where variability and customization become important. With a system in place on top of an SOA, a service provider could potentially dynamically deliver thousands of application variants to end users, in the context of different syndication partner sites, just as easily as it could syndicate out static content in the form of ads. The difference is that the application variants are being "morphed" on the fly to represent different presentations, logical flows, and back-end interactions, whereas an ad is just a static piece of data.

Syndication is the newest application delivery mechanism that necessitates a real-time, on demand delivery capability for providing mass customization.

Summary

Service-oriented architectures lay the groundwork for enabling a new level of reuse and customization from a relatively small set of backend services. However, SOAs alone have limitations and cannot solve the problem developers face to deliver many different variations of an application to support different types of functionality and behavior. This problem is compounded by the new need to syndicate applications out into large numbers of partner sites in order for service providers to extend their reach to a broader audience.

SOAs are a first step towards realizing faster development with fewer resources. In order to realize the ultimate goal of on demand real-time delivery of finished applications, an automation component needs to be in place on top of the SOA that performs the fabrication and delivery of applications in the contexts where they are needed.

Mission-critical data and Web services deployment

Automated Identity Management

BY TONY NADALIN

Tony Nadalin, distinguished engineer and chief security architect for IBM Software, is a passionate technologist and coauthor of the Web Services Security (WS-Security) standard. In this role he also serves as the primary security liaison with Sun Microsystems and Microsoft Corp. for IBM. As such, he has facilitated the collaboration of vendors and industry bodies alike.

In this Q&A, Tony Nadalin, chief of security architecture for IBM Software discusses ways in which companies can protect their mission critical data even as they connect their networks with outside partners.

What Is Identity Management? Why Is It Important to Web Services?

The goal of identity management is to use accurate knowledge of your users to reduce costs, improve security, and achieve regulatory compliance. It's a broad and multifaceted discipline that has rapidly emerged to become a top IT initiative. Objectively understanding your priorities is a critical factor in planning an integrated identity management solution, especially in light of deploying Web services.

In terms of cost, today's IT landscape has shifted from the computer being the costly resource to the actual users being the costly resource. As members of the technology industry, we need to work to reduce manual labor to bring these costs down. We can do this through automation.

Besides Reducing Manual Process, What Are Some Other Business Benefits?

The fundamental business benefit of deploying a Web service is the connection or interoperability of disparate networks, allowing for more efficient business process flows. However, making sure the networks have consistent security policies can be an administrative nightmare, especially when the networks are owned and operated by separate entities as business partners. It is vitally important to establish a consistent identity management strategy that gets absolute buy-in from end users, the IT staff, and the management team.

What Steps Should Organizations Take to Implement a Consistent Identity Management Strategy?

An important dimension of the solution is its incremental implementation; the specific order depends on your organizational needs. The three steps to this are to 1) assess the needs of your organization, 2) identify security policies like who gets access to specific data and who controls certain access points, and 3) establish a business case and run the ROI figures. When you consider that help desk personnel currently spend up to 30% of their time resetting passwords, man hours saved can be huge.

Let's Say You Want to Expand Security Policies to a New Web Service You Are Deploying. What Do You Need to Keep in Mind as Some Unique Characteristics of This Type of Environment?

You need to make sure different applications can access and share information across systems. And the way you do this is by implementing an autonomic identity management strategy that takes advantage of open standards and APIs.

By using an open standard like J2EE, network administrators are able to extract security policies directly from an application, user profile, or data stream. It takes this directly from the packet container and centralizes this information where it can be accessed by a security administrator via an access management solution.

J2EE allows these security policies to be shared throughout the Web services network regardless of where the network lies. This ensures that security policies are consistent across all networks within the Web service.

Most APIs can be accessed by J2EE in this way.

What APIs Can't J2EE Access and Where Do They Sit?

The two major exceptions are the CallerInRole API and the getCallerPrincipal API. CallerInRole indicates whether the application has permission to perform a particular task. getCallerPrincipal indicates what application a user is currently using. These two APIs are supported by every J2EE implementation, which guarantees interoperability between vendors. For example, a client will be allowed to perform a particular task only if the principal associated with the client call has been assigned by the deployer and the client has permission from the principal to perform the task. The container provider is responsible for managing and enforcing the security policies at runtime and for providing the tools used by the deployer to manage security during deployment.

How Is This Relevant to Securing Applications?

J2EE helps access these APIs regardless of platform. In a heterogeneous network it is important to be able to collect this information. The alternative is for vendors to provide APIs. However, adding APIs creates more manual configurations and actually has a negative impact on performance and application portability.

Policy Decision Points (PDPs) collect and centralize policy information. For example, a WebSphere deployment will have deployment, configuration, and runtime policies, and all this information needs to be consistent across the entire infrastructure relating to a particular Web service. PDPs determine where to find this information, then collects and retrieves it.

Is There a Standard Framework That IBM Recommends When Expanding Security Policies to a WebSphere Environment?

Yes. Identity management plays a more prominent role in managing a Web services environment or a service-oriented architecture, and there needs to be a standard set of practices that organizations follow to ensure security policies are implemented in the deployment stage and integrated across the entire organization.

First there needs to be a foundation layer that stores security policies and synchronizes data in the security directory. Then there needs to be a control layer that acts as the traffic cop, providing applications and user access to information. The third layer is the identity lifecycle layer, which manages things like policy management and provisioning.

The last layer is the federation layer that allows companies to share identities securely, giving employees, customers, and partners access to systems and resources throughout the supply chain.

What is Federated Identity Management? How Will Java Environments Benefit From It?

Federated "schemes" are intended to allow for a single sign-on experience within a rapidly evolving heterogeneous environment, that is, an environment made up of many distinct enterprises. An enterprise has its own user registry, maintains relationships with its own set of users, and has the means of authenticating these users. A federated scheme allows users in one enterprise to leverage relationships with a subset of enterprises through participation in a federation, granting users access to resources at any of the federated enterprises as if they had a direct relationship with that enterprise.

Existing single sign-on solutions are limited to an environment in which there are pre-established business agreements between the participating enterprises. These business agreements are used (in part) to establish trust and to limit and define how information is transferred in a secure manner. These business agreements also include technological agreements or rules on how to translate user identities from one enterprise to another, and how to transfer the information used to vouch for these users across enterprises. Java environments like J2EE will benefit by being able to incorporate a ubiquitous architecture for federation.

There Are so Many Standards Out There. Which Ones Do Organizations Need to Support?

Just like in the business world, IT environments at different companies speak different languages; however, it remains important to be able to share information across those language barriers. Currently, there are three standards that customers are actively using: WS-Federation, Liberty, and SAML. Customers should look for a federated identity management solution that supports all three and easily ties into their existing identity management infrastructure.

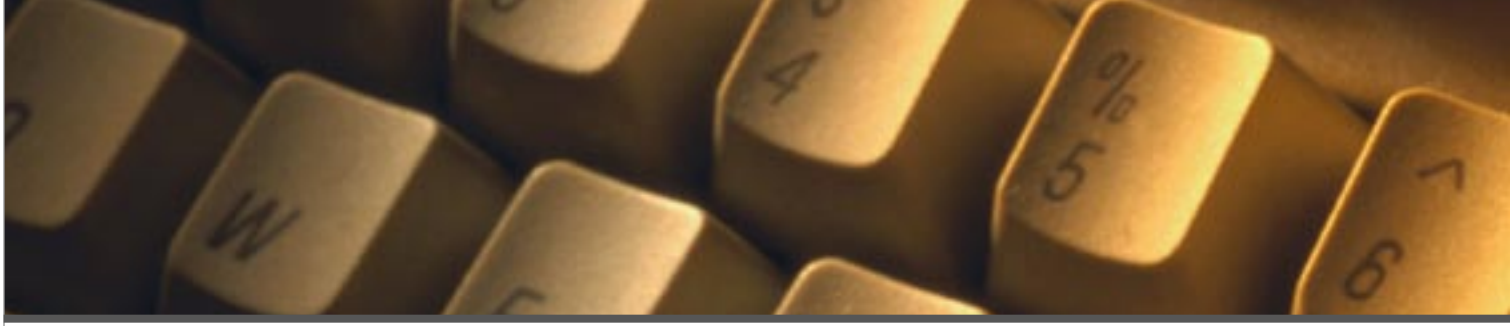
How Does This Relate to IBM's Vision of the On Demand Business?

Model-driven development and management of secure applications and solutions is emerging as a key concept in meeting the demands of an on demand enterprise. In a given enterprise, there are various people who contribute to the modeling, development, deployment, and management of the security and privacy aspects of a business application. For example, the container provider I spoke about previously is responsible for enforcing the security policies at runtime, providing the tools for managing security at runtime, and providing the tools used by the deployer to manage security during deployment.

You need to consider the life cycle of an on demand business solution, and provide an approach as to how the security and privacy policies can be modeled using policies and rule templates and attached to business process and models. It can also be designed and implemented through container-managed or application-managed environments based on the modeled artifacts. You can deploy directly into an infrastructure and customize to meet the security and privacy needs of the consumer. These can be monitored and managed to reflect a consistent set of policies across the enterprise and layers of application infrastructure.

How Can Someone Tie These Security Issues to Business Goals?

They need to use a pragmatic approach to finding intersection points between a platform independent of security and privacy policies, and the concrete articulation of policies. We need to use business vocabulary in IT so that business policies are taken into account during the IT deployment stage. This type of approach offers a way to integrate monitoring and management of the IT infrastructure to business policies.



Using the Search Capabilities of WebSphere Portal v5 Part 2

A step-by-step process

BY TILAK **MITRA**

Tilak Mitra is a certified IT architect in AIS, EAD Southeast sector in the Ft. Lauderdale/Miami area for IBM. He specializes in mid- to large-range enterprise and application architectures based on SOA, J2EE, MQ, and other EAI technologies.
tmitra@us.ibm.com

The first article (Vol. 3, issue 9) in this series on WebSphere Portal Servers (WPS) search capabilities introduced you to some basic capabilities of the native search features of WPS.

In this article we will use a hypothetical scenario in which we configure a portal search portlet with user-defined categories. We will create user-defined categories in which the search results will be contained. Once this portlet is created, installed, configured, and run, we will be able to see the results of the search displayed in one or more categories that we will create.

We will be using the same IBM internet URL (www.ibm.com) that we used in part 1 as the site to crawl for content.

Setting Up the Collection

In this article we will customize the collection that we are going to create. We'll begin with a brief overview of the process involved in setting up the collection and then walk through the step-by-step process.

Process Overview

A collection is defined as a set of URLs that are crawled to gather contents. The URLs can be on either the Internet or an intranet. It is immaterial where a particular URL points to as long as the content is reachable for Web crawling by WPS' Search Engine. The content that is crawled by the search engine needs to be categorized according to a taxonomy using categories defined by the user. The categories are hierarchical, with the ability to create subcategories to any practical level of nesting. The nodes in the category tree need to be associated with rules. These rules control the type of information that is retrieved from sites. The retrieved content complies with the given rules in order to be inserted into the collection and made available for search. Once each node, or leaf, in the category tree has an associated rule, the configuration for the collection is complete.

With the necessary configuration steps for the collection complete, the actual collection process can be triggered. Once all of the content is retrieved, each item may be edited before being either accepted or rejected; accepted items will be presented as search results to the user.

A Step-by-Step Process

1. Log on to the portal as a portal administrator (e.g., wpsadmin).
2. Go to Administration >Portal Settings > Search Administration. Select Create Collection.
3. Supply the following information:
   - *Location of the Collection:* IBMSearchUserDefined
   - *Specify Collection Language:* English
   - *Select Categorizer:* User-Defined (This states that we are not going to use any of the out-of-the-box categories but will instead create a user-defined one.)
   - *Select Summarizer:* Automatic
   - Check "Remove common word…"
   - Click OK.

4. Now we will define the taxonomy and the category tree. Click on Category Tree.
   - Change the default top level category name (root) to Products. Click Rename.
   - We will be creating two subcategories of "Products" as an example to demonstrate categorization of IBM products into two domains – software and storage. Enter a new subcategory called software. Click Create.
   - Repeat the same step and create another subcategory under products and name it storage. (*Note:* While you can create any level of nesting for categories and sub-categories, for our example we will create two subcategories under the top-level node.) The category tree should now look like Figure 1.

5. Each of the subcategories needs to be associated with a rules filter. A rules filter essentially filters the incoming documents based on a set of keywords. Only documents that satisfy those keywords may be added to a given category. A rules filter has a name and list of keywords based on which the filtering rules are applied. Rules may be applied based on the URL text or content (inside the URL). For the sake of simplicity we are going to apply the filtering rules based on the URL text.
   - To set up the rules for the software category, highlight software and click Create from the Manage Category Rules Software panel. Input the values as shown in Figure 2 and click Create.

– Click OK.
– Repeat the same steps for the storage category using the values as shown in Figure 3.

6. Click the Back application button to complete the task of creating the category rules.
7. Now we will define the site to crawl and apply categories and filtering rules to the documents referenced from the site. Click Add Site in the Sites in Collection: IBMSearchUserDefined panel.
8. Supply the general parameters as shown in Figure 4 and click Create.
9. Now we need to edit the site and add the categories that we defined. This will ensure that the incoming documents will be a part of the categories after the filtering rules have been applied.
   – Click Edit Site.
   – Highlight the software link and click Add to List.
   – Perform the same operation for the storage link.
   – Once the Destination Categories list looks like Figure 5, click OK.

10. A time schedule defines how frequenly the collection will be updated. This ensures that the content of the collection will be refreshed periodically from the actual (external or internal) site. To set this up click Create on the Collection Update Schedule panel. Enter a starting time followed by a time interval based on which the collection content will be updated. A sample schedule is shown in Figure 6.
11. You can force the initiation of the collection creation process by clicking the Start Collecting link in the Sites in Collection:IBMSearchUserDefined panel. Until the Site Status shows a completion status for the document collection process, no other activities should be done in this administration screen.
12. Referring to Step 8 above, the Add all Documents to Collection Automatically attribute is not checked. This implies that the incoming documents (from the site) are not automatically added to the collection. We need to look through all the Pending documents and determine which documents to accept, modify or reject.
   – Click on the Pending Documents link under the Manage Document Collections panel.
   – Click on Reject for each of the documents that are in the collection till the documents as shown in Figure 7 are the only ones left.
   – For the fourth, fifth, and seventh documents, as shown in Figure 8, click the Edit icon and make sure the only category present in the Update Destination Categories section is Products-Software.
   – Repeat the same process for the first, second, and eighth documents, but this time make sure the only category present is Products-Storage.
   – For the third and sixth documents, both categories Products-Software and Products-Storage should be present.
   – Now click the Apply radiobutton in order to complete the process of accepting the documents.
   – Finally, click Cancel in order to complete the process of accepting, rejecting, and editing the pending documents.

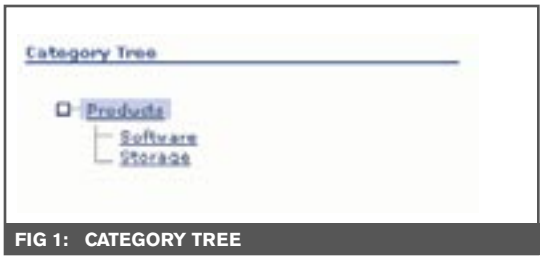


FIG 1: CATEGORY TREE

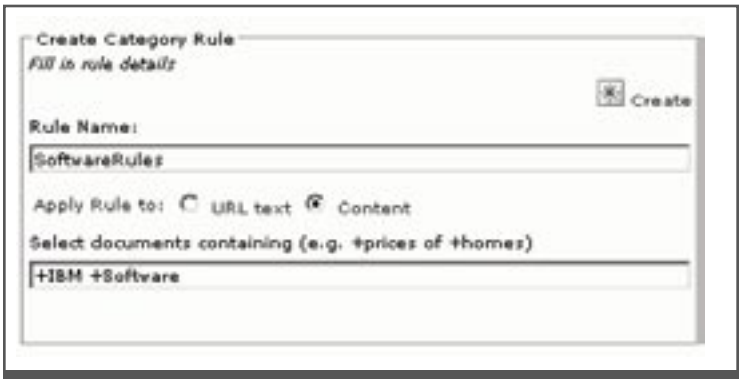


FIG 2: CREATE CATEGORY RULE

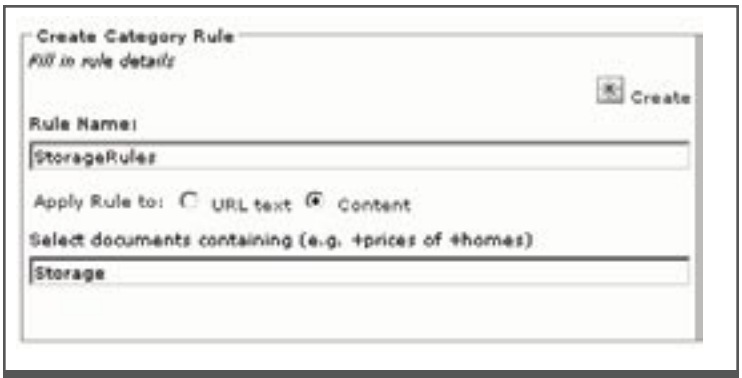


FIG 3: CREATE STORAGE RULE

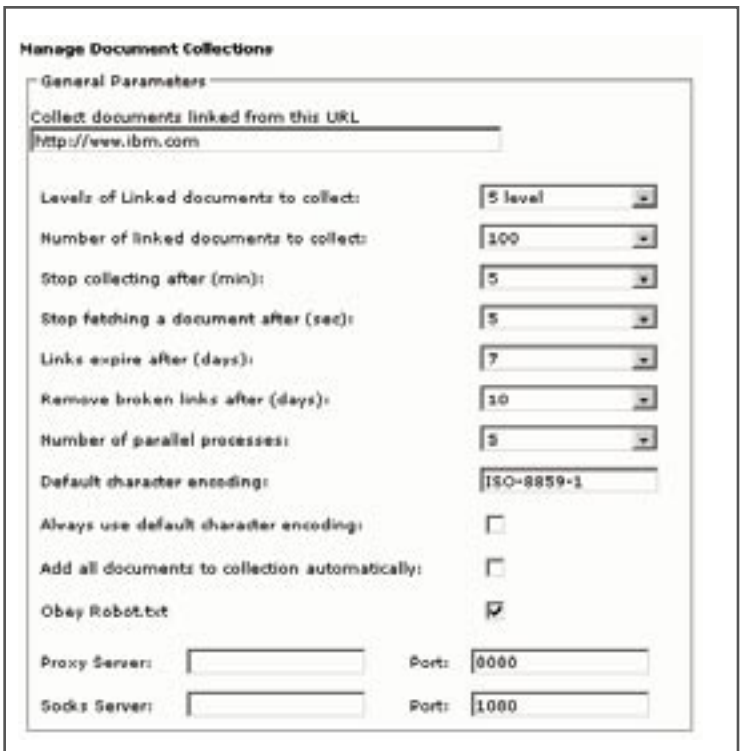


FIG 4: CREATE GENERAL PARAMETERS

FIG 5: ADD DESTINATION CATEGORY

FIG 6: CREATE SCHEDULED UPDATES

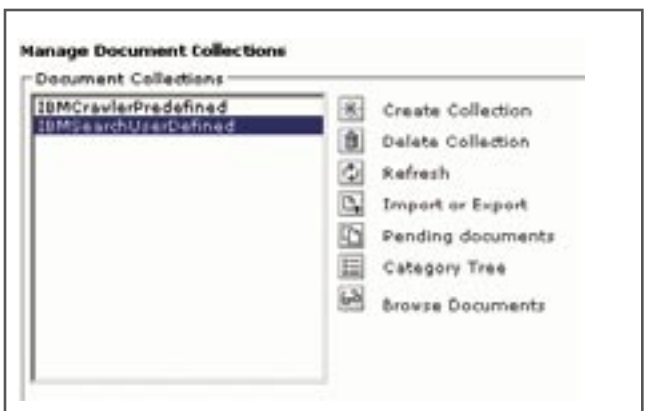


FIG 7: PENDING DOCUMENTS TO BE EITHER ACCEPTED, MODIFIED, OR REJECTED

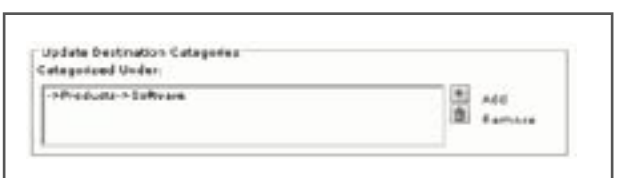


FIG 8: EDIT DESTINATION CATEGORY

FIG 9: FINAL SEARCH RESULTS

Configuring the Search Portlet

This section is very similar to the same section in part 1. Referring to the steps in part 1 as-is, I will mention only the parameters and attribute values that are different.

1. As in part 1.
2. The IndexName attribute should have the value C:\WebSphere\AppServer\IBMSearchUserDefined. (This is how the association is created between the custom portlet and the index that is created in the previous section.) All other attribute values remain the same.
3. The Title attribute value is Custom Search On IBM Internet.
4. As in part 1.
5. As in part 1.

Adding our Custom Portlet

This too is very similar to the same section in part 1. Referring to the steps in part 1 as-is, I am going to mention only the parameters and attribute values that are different.

The only difference is that the portlet Custom Search On IBM Internet needs to be searched for before being added to the My Work page in the portal.

Running the Search

Back in the My Work top-level page we can see that our portlet has arrived, ready to be executed. In the search text field, I keyed in the word tape and hit the search button to the right. An excerpt of the output of the search results is shown in Figure 9.

Notice that each search result is categorized using the categories that we created and configured in this example.

Try searching with other keywords (e.g., Lotus, disk, DB2, storage, etc.) and see how the search results change. Pay special attention to the categories for each of the search results and you will see how they are displayed under the categories that we defined.

Conclusion

This article showed you how to create custom user-defined categories and subcategories. We also learned how we can apply filtering rules to the incoming document content to pick and choose which documents to accept, modify or reject before they are added to the collection.

This feature adds a lot of flexibility in creating a rich and extensible collection of documents that can be indexed for efficient searches. All of this comes right out of the box with WPS and is always a very handy feature to look into while considering a search engine implementation within WPS, with minimal effort!

Resources

- *The WebSphere Portal InfoCenter*: http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html.
- The IBM *Redbook:* www.redbooks.ibm.com/abstracts/sq246076.html?open
- *The WebSphere Portal Zone*: www-106.ibm.com/developerworks/websphere/zones/portal/.

Achieve higher quality

in less time, with fewer resources.

Parasoft® Automated Error Prevention (AEP) Products:

Specifically designed for under-staffed, over-committed development organizations like yours.

Parasoft AEP Products ensure that comprehensive error prevention practices are applied consistently and uniformly across your entire team.

By automating compliance to a unified set of testing and coding standards, Parasoft AEP Products enable every team member to follow the same best practices and the same error prevention techniques – including coding standards analysis, unit testing, code review and regression testing.

Automated testing and standards compliance for any team configuration.

Parasoft AEP Products automate error prevention for Java, C/C++, Web Services, the Microsoft® .NET Framework, Embedded Development, Web Development, Database Development and more.

For details go to: www.Parasoft.com/AchieveQuality

Call: 888-305-0041 x3307 Email: AchieveQuality@Parasoft.com

PARASOFT®
We make software work.™

Founded in 1987, Parasoft's clients include IBM, HP, DaimlerChrysler and over 10,000 companies worldwide.

Availability

Parasoft AEP Products are available on Linux, Solaris, and Windows NT/2000/XP.

Contact Parasoft for details and pricing information.

Parasoft Corporation
101 E. Huntington Dr., 2nd Flr.,
Monrovia, CA 91016

Electronics companies face the challenge to get up to speed

Innovate and Deliver

BY GUY **BARLOW**
& CHRISTOPHE **BEGUE**

Guy Barlow is the Software Group Electronics Marketing Manager at IBM.

Christophe Begue is in Industrial Sector Solutions Development in IBM Software Group

Whether they produce cameras, equipment to make semiconductors, plasma TVs, medical equipment, printers, or cell phones, electronics companies share an all-encompassing requirement; they must continually bring more innovative products to market faster than ever before.

This is no small challenge. Electronics companies must innovate and deliver at a relentless pace despite having long lead times for product development and little control over what and when consumers want to buy. The longer it takes to get to market, the more their profits erode as a result of intense competition and constantly evolving technologies in their products.

Electronics companies also face challenges related to their internal structures. Some haven't fully recovered from the technology sector downturn. Many have expensive and inflexible cost structures or their product development and supply chain organizations and data are isolated.

They also face customer cynicism about the buying experience and product value as well as increasing regulatory demands. They must improve their operations and cut costs.

Many electronics companies have "looked at their businesses and determined that to be successful in the future, they need to develop new opportunities within and beyond their existing value chain," said Guy Barlow, marketing manager of the electronics industry within the IBM Software Group. As a result, new business models are emerging. For example, some electronics companies are challenging the entertainment industry, saying its inflexible stance on digital copy protection threatens technological innovation and the public interest. Another example is Apple Computer, which is moving into the music business with its iTunes Music Store.

To tap into new opportunities in the value chain, enter new businesses and markets, cut costs, and compete at the industry's make-or-break pace, electronics companies need to integrate their business and IT processes within and beyond their own organizations. They need to become on-demand enterprises – integrated across all groups and partners, allowing them to respond with speed to customers' and suppliers' demands, competitive threats, and regulatory requirements.

As an electronics company evolving into an on-demand enterprise itself, IBM is working closely with the large range of other electronics companies to address their critical challenges and help them make the transition. Currently, more than 75% of electronics companies use IBM products and 17 of the top 20 use IBM WebSphere products. In addition, IBM has a network of over 170 business partners for electronics solutions worldwide.

This experience with electronics customers guided IBM's creation of middleware solutions that are designed specifically to address the industry's most pressing challenges. The solutions are part of IBM Software Group's effort to deliver industry middleware solutions – a strategy based on customer purchasing behavior that indicates they prefer to buy solutions designed for their industry. Each solution contains four optional elements: IBM middleware technology from its five software brands (WebSphere, DB2, Tivoli, Lotus, and Rational), IBM's industry-specific middleware, industry-specific services expertise from IBM and others, and industry-specific application software from IBM's network of ISVs.

"We designed the solutions based on what electronics customers had done and what IBM had done to address these specific industry challenges," said Christophe Begue, industrial sector solution development, IBM Software Group. "We knew we also had to build on their existing infrastructures with open standards and take an approach where the client could start by addressing any piece they chose."

Begue's team's work with electronics customers brought several things to the forefront. "We saw that their supply chain was a good place to cut costs and could benefit from collaboration and innovation management capabilities," Begue said. "So we focused on value chain and product life cycle management." Four middleware solutions were then created for the electronics industry.

Electronics companies need to improve efficiency in their supply

chain processes. Many have hundreds of global partners with whom they must coordinate people, processes, and information. They also need to manage fluctuating lead times, inventory, and unanticipated changes to supply and demand.

The IBM Middleware Industry Solution for Electronics Value Chain Management provides solution assets in the areas of supply chain collaboration, RFID, indirect procurement optimization, business performance management, and regulatory compliance. The solution helps companies reconcile demand, supply, and inventory and align financial responsibilities with supply agreements. It helps them standardize supplier and customer processes to speed up execution and information sharing. The result is adherence to industry standards, reduced operational response times, and increased visibility throughout the partner community.

For example, a leading consumer electronics manufacturer based in Asia needed to reduce its inventory; costs and production lead times and wanted to eliminate the manual invoice process. The IBM solution enabled the company to implement vendor-managed inventory and Rosettanet industry standards supply chain collaboration processes. With the IBM solution, inventory costs were reduced by 50%, production lead times by 20%, and customer satisfaction, loyalty, and market share also improved significantly.

The IBM Middleware Industry Solution for Electronics Sales and Service is a solution to address a manufacturer's end-to-end customer service and sales requirements for the life cycle of a product. This solution includes remote diagnostics and customer support call centers; authoring tools to generate product service, parts, and testing content; configuration management; telematics and Web-based service documentation and parts systems.

In addition, the solution includes CRM, portals and wireless, e2Home and Broadband, Sell & Support offerings as well as integration and collaboration services to support these offerings. The end result is a solution that helps clients improve and transform their marketing, sales, and service business processes including contact centers, field services, and spare parts logistics.

"Electronics companies need to integrate their business and IT processes within and beyond their own organizations"

The IBM Middleware Industry Solution for Electronics Product Lifecycle Management helps companies improve product design and engineering efficiency. It helps them deal with complex and geographically dispersed collaborative design, lack of integration in key engineering systems, and the need to shorten development cycles. It does so by synchronizing outsourced product design data with internal systems and enables collaboration across engineering groups using virtual design models. This improves design productivity and quality, provides the ability to leverage outsourced designs, and speeds development. Additionally, the solution focuses on the embedded systems' life-cycle management area and helps companies improve the development of software systems. Today's software requires complex design processes for development and maintenance. This solution provides a development framework that manages these requirements by standardizing tools and processes. The result is a shortened development cycle across multiple platforms and locations.

One example of an IBM customer using this solution is a global provider of power and automation technologies that enables utility and industrial customers to improve performance while lowering environmental impact. With operations in 100 countries and nearly 115,000 employees, the company needed to upgrade its manual, inefficient design systems and integrate disparate engineering and business applications. It used IBM middleware and ISV software to integrate databases and the result was a 30% reduction in product development time, a 70% improvement in mechanical design quality, and a 40% improvement in bill of materials.

The IBM Middleware Solution for Electronics Industry Production Solutions addresses the production management process from sourcing to fulfillment by connecting the plant floor, enterprise, and supply chains through the use of collaborative manufacturing infrastructure, MES/SCM/ERP and PLM Solutions, and systems integration. It includes elements of business process design with lean manufacturing techniques. Furthermore, ERP integration with MES and eDiagnostics are part of the solutions. This solution enables manufacturers to increase the speed of product development, manufacturing ramp-up, and cycle time along with decreased manufacturing costs and improved asset management.

This integration and streamlining of processes and information will be critical to electronics companies trying to compete as their extremely dynamic and fast-paced industry continues to heat up. "Electronics companies have always had to cope with a continuous need to innovate and innovate very quickly while knowing that prices and profits will drop if they're too slow to market," said Barlow. "These solutions give them a way to address their challenges and be more responsive to an ever-changing market."

A useful set of tools for preventing bugs from infiltrating software before deployment

Parasoft WebKing

BY JAY **JOHNSON**

Jay Johnson is a consultant for the WebSphere software group at IBM.
jay2ee@hotmail.com

Quality-conscious developers are familiar with the idea of coding checklists. The code you write must measure up to all the criteria on the checklist, from "no grammatical errors in the comments" to "performs all required functions." Based on these checklists, we have code reviews. A good code review takes time, but is certainly worth the effort. Such reviews can prevent many costly errors. However, when crunch time hits, thorough code reviews are often impossible. That's where a tool like Parasoft's WebKing can help.

For several decades tools to automatically generate and run tests have been available. As I wrote in *Program Smarter, Not Harder*, automated testing tools can provide the most bang for the buck in software development process improvement. After years of fighting software wars, developers have figured out that catching errors using static analysis relatively early in a development cycle rather than during user acceptance testing can mean the difference between project success and failure. These days, therefore, the biggest potential for process improvement may be in the concept of automating error prevention.

PARASOFT
401 E. Huntington Dr.
Monrovia, CA 91016
Phone: (888) 305-0041
International: (626) 256-3680
E-mail: info@parasoft.com

Static Analysis

Static analysis provides a code review by automatically inspecting source files to pinpoint code that could reduce application functionality, accessibility, and usability, or impair transformations and updates. As illustrated in Figure 1, clicking the Static Analysis button launches the analysis by applying the default or customized set of static analysis tools.

Static analysis uses tools that verify whether:

- Files contain broken links and navigational problems
- HTML, CSS, JavaScript, and ASP/VBScript code complies with W3C standards and industry guidelines
- HTML and CSS code complies with Section 508 and W3C WAI accessibility guidelines
- XML is well-formed and valid
- HTML and XML files contain typos and misspelled words

In addition, static analysis can verify unique requirements that your team wants to check. For example, if your team needs to satisfy custom coding requirements, project-specific design requirements, and project-specific content requirements, you can automatically generate a rule that verifies whether each requirement is satisfied, and then check compliance during static analysis. As the code is analyzed, errors along with recommended fixes, are displayed.

You can customize static analysis tests by:

- Enabling or disabling any of the built-in tools available for static analysis
- Customizing the parameters of any tool applied during static analysis.
- Designing tools that verify compliance with unique team or project requirements, then configuring WebKing to apply those tools during static analysis
- Running static analysis on selected files or directories
- Running static analysis on the click paths you record and generate for functional testing
- Configuring WebKing to suppress error messages that are not relevant to your project

The single most powerful feature, however, is the ability to invoke WebKing's static analysis API from Java, JavaScript, or Python, as illustrated in Figure 1, thus making it possible to fully automate a code review. Whether you use WebKing or adopt another approach, whatever effort you put into creating an automated code review will be repaid manyfold in error prevention.

Functional Testing

Static analysis can make a big difference, but preventing bugs from showing up in deployed software requires automated functional testing. While static analysis finds problems by inspecting source code, functional testing finds problems by simulating how click paths would

operate in a browser. Functional testing involves verifying whether designated click paths execute correctly. Functional testing exposes problems such as:

- JavaScript runtime errors
- Pop-up windows, page changes, and other effects that do not work as expected
- Frames that do not load correctly in the context in which they are used
- Valid pages that are blank or incomplete
- Server-side program crashes and exceptions
- Server errors and failures
- Unexpected page content changes
- Unexpected click path flow changes

WebKing facilitates functional testing by providing ways to generate and record test click paths without requiring any scripting. You can run functional tests by clicking the Test Functionality button and then use the Path Creation wizard to create the initial paths (see Figure 1). This wizard allows you to automatically generate paths or record specific paths by clicking through the application in a browser. WebKing then executes the initial paths and reports errors if it cannot execute one of the recorded path items, or if a runtime error occurs while executing the path. You can then extend this initial set of test paths in at least two ways:

- Automatically record your application's most popular click paths by prompting WebKing to analyze your server log files.
- Use the Path Creation wizard to automatically generate a variety of unique paths and/or to help you record specific click paths that correspond to your specification and use cases. By combining these options, you can quickly represent your most critical and popular paths as well as an array of other potential paths that might otherwise go untested.

Once you have verified that one or more paths work correctly, you can automatically record path execution results to establish baseline regression controls. After these regression controls are established, WebKing compares all subsequent path execution results to the baseline results, and then reports unexpected path flow and content changes. WebKing performs regression testing at the code level, so it exposes even subtle changes that are not detectable at the GUI level (for example, minor changes to scripts or hidden form options). It reports the differences both textually and graphically to help you pinpoint and understand any unexpected changes that occur. In addition, you can increase the scope and comprehensiveness of your functional testing by dynamically populating forms with values stored in a data source.

There are a number of ways to customize functional testing using WebKing, including populating forms with values stored in a data source, or recording and generating additional paths via the Path Creation wizard and GUI controls. Another useful feature is the ability to generate a site's most common paths by prompting WebKing to analyze its server log files. You can also extend the existing paths via JavaScript.

WebKing can be configured to run a script or tool every time it accesses a designated path step – for example, to perform initialization, set or remove a cookie, etc. It can also be set up to create regression controls for pages returned after forms are populated with data source values and then submitted.

After configuring WebKing to populate a path's forms with data source values, it will dynamically populate the designated forms every time it accesses that path. In addition, if you configure a test suite that exercises the path, WebKing will automatically populate and execute that path once for every available combination of data source rows. Consequently, if the path's forms use values from a single data source, that path is executed one time for every row in the data source. If the path's forms use values from multiple data sources, that path is executed and tested one time for every possible combination of rows. As a result, you can create and test a large number of path instances with relatively minimal effort.

Load Testing

Load testing involves exercising your application with virtual users and verifying whether it supports the tested traffic levels, patterns, and combinations. It is typically used to verify whether the application will perform adequately under the expected loads and surges, determine system capacity, and identify the source of any bottlenecks.

WebKing facilitates load testing by providing "intelligent" virtual users and sophisticated ready-to-run load test scenarios. To run a load test, you can click the Load Test button, and then choose one of the four preconfigured load test scenarios (bell curve, buffer test, steady load, or linear increase) that are included with WebKing. This initial test then simulates the selected load pattern and virtual users generate traffic by automatically generating paths through the application.

If you want to verify how your application performs under its typical load distributions and traffic combinations, you can prompt WebKing to automatically generate a custom load test based on your application's actual usage patterns.

Tell WebKing where to locate your server log files and it creates

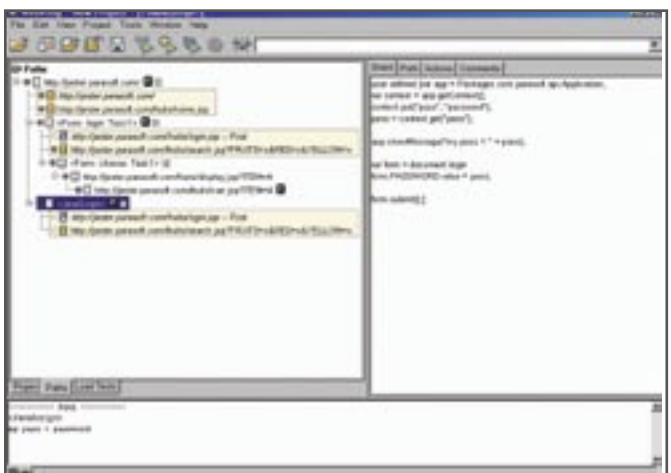


FIG 1: ANALYSIS PATHS AND A CUSTOM STATIC ANALYSIS SCRIPT

a test scenario that simulates your application's actual load distributions and your users' actual path choices. In addition, WebKing allows you to distribute virtual users across remote server machines to simulate extremely large loads and/or test from different locations.

By placing machines at strategic locations inside and outside the network, you can perform stress testing on specific pipes into the system and determine the capacity of each pipe, as well as the total system capacity. For example, imagine that you want to determine the source of a reported bottleneck.

First, you might configure several WebKing server machines immediately behind your Web server in order to determine the capacity of that link into your application. You would then use the WebKing client to design a test that ran a linear increase of virtual users on these machines.

By running the test and watching the result, you could determine if and when the increased number of virtual users caused performance degradation. You could then apply this same strategy to determine the capacity of each individual link to the application, and move progressively farther from the Web server until you were testing from remote locations far outside the network.

You can use the Confirm Scenario Wizard to select which scenario you want to use. Consider the following points when making the selection:

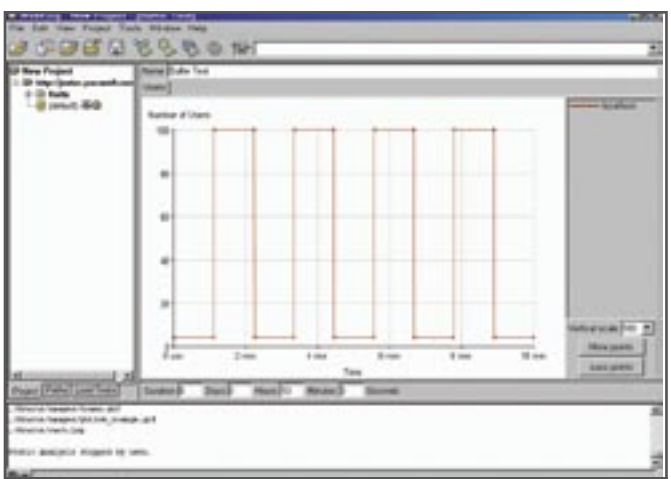


FIG 2: A BUFFER TEST TO DETERMINE HOW RESOURCES ARE ALLOCATED AND DEALLOCATED

- The Bell scenario simulates the typical daily user distribution (starts at the lowest point immediately after midnight, rises gradually in the morning, peaks towards the middle of the day, declines gradually in the afternoon, then returns to the lowest point at the end of the day). This is useful for expected usage testing, which determines how the site handles normal load patterns.
- The Buffer Test scenario (illustrated in Figure 2) simulates a variable load over time. This helps determine whether resources are released when the load decreases. It also helps you determine whether overall performance degrades over time.
- The Linear Increase scenario simulates a load that increases linearly over time. This is useful for stress capacity testing, which helps you determine how many users your application can support.
- The Steady Load scenario simulates a steady load of users over time. This is useful for endurance testing, which helps you determine whether performance degrades over time.

WebKing will run a load test based on the specified scenario settings as well as the related virtual user profile settings and machine settings. Using default settings, WebKing will run the test on the local machine and create virtual users that create unique paths through the application.

Results are reported in a variety of customizable formats. When the test completes, WebKing will show a test summary in the right panel. To access results in bar graph, text format, or histogram format, choose the appropriate command from the Views box. These results can be sorted, filtered, and customized.

You can customize load tests by:

- Prompting WebKing to automatically generate a custom load test scenario that simulates your application's actual load distributions and your users' actual path choices (based on log file analysis)
- Customizing the paths virtual users follow during the test (virtual users can follow specific paths recorded or generated for functional testing, or they can automatically create original paths through the application)
- Customizing the tools virtual users apply to verify functionality under load
- Customizing virtual user browsing characteristics such as delay between clicks, types of links followed, cookie use, etc.
- Configuring WebKing to perform stress testing by running tests from multiple server machines located at different locations within and without the network
- Customizing the number and distribution of virtual users over time and over machines
- Customizing the type and distribution of path combinations over time and over machines
- Customizing the test duration
- Designing a filter that focuses results on the details most important to you
- Using Windows monitors and SNMP monitors to collect network information and system performance data during load testing
- Using WebKing Windows services so you can monitor, start, stop, and synchronize load tests on remote server machines from a client machine – without having to manually start WebKing on each server machine
- Using "Do not parse HTML" mode to generate a much higher load on your server with fewer machines running WebKing (when generating load is a higher priority than functional testing)

Summary

Parasoft's WebKing runs on Windows, Linux, or Solaris; it can be a very useful set of tools for preventing bugs from infiltrating software before deployment. It has an impressive static analysis capability, flexible functional testing features, and relatively easy-to-use load testing facilities.

From Chapter 4

Build and Deploy Procedures

BY ROLAND **BARCIA**

Roland Barcia is a consulting IT specialist for IBM Software Services for WebSphere in the New York/New Jersey Metro area. He is the author of one of the most popular article series on the developerWorks WebSphere site, www-106.ibm.com/developerworks/websphere/techjournal/0401_barcia/barcia.html, and is also a coauthor of *IBM WebSphere: Deployment and Advanced Configuration*. You can find more information about him at http://web.njit.edu/~rb54
barcia@us.ibm.com

This chapter, "Build and Deploy Procedures" is excerpted from the new book, *IBM WebSphere: Deployment and Advanced Configuration*, authored by Roland Barcia, Bill Hines, Tom Alcott and Keys Botzum. © International Business Machines Corporation 2005. ISBN 0-13-146862-6. To learn more, www.phptr.com/title/0131468626.

In the last chapter, we provided you with a quick start to WebSphere Application Server (WAS) by configuring and deploying the Personal Trade System. The deployment was simple and straightforward, using the WAS administrative console to do most of the work. With small applications, this approach is often sufficient. However, for most J2EE applications, deployment requires much more planning and process. In fact, the entire application lifecycle requires rigorous processes.

In this chapter, we will focus on three closely related areas: builds, deployment, and server configuration. We will start by discussing some key reasons why rigor is important and then define some terminology that we'll be using for the rest of the chapter. Then we'll identify three common approaches to build and deploy. Following that, we will focus on automation.

Automation of the build and deploy steps can reduce the amount of effort required to produce and then deploy an application. Automation can also reduce the likelihood of human error—not an insignificant concern in large and complex projects. In addition to automating the assembly and building of applications, we will offer advice on how to automate server configuration.

Procedures Matter

The enterprise development model is more complex than traditional small application development—code must be written, compiled, assembled, and then deployed (presumably with some testing included). That is, after the code is compiled, the application must be assembled from a set of modules into a single application EAR. In addition, deployment includes the implicit concept of "binding," which is taking a built application and binding its contents to the target runtime environment. These two concepts are very powerful and give applications a great deal of flexibility, but they do require additional discipline. Build and deploy in these situations are core functions that must be carefully managed like anything else.

Setting up high-quality procedures correctly can set the tone for the rest of the project.

In our experience, failing to develop solid procedures often will cause projects to experience serious problems and occasionally will result in project failure. In this book, we are assuming a rigorous approach to development, roughly like this:

- Developers write code following high-quality software engineering principles. This includes a configuration management system for their source code and other artifacts.
- Developers execute unit tests against their code before checking it into the repository.
- Code is built into a complete system following a well-defined procedure.
- Code is deployed to a runtime environment (an actual WAS instance) following some well-defined procedure.
- Deployments are verified using a well-defined set of component test cases. Component tests are essentially a subset of development unit tests and are focused on cross-component function and environment sanity.
- The target application server environment is configured using well-defined procedures.
- Multiple runtime environments are available for different purposes (development integration, system test, performance, etc.) as defined in Chapter 16, "Ideal Development and Testing Environments."

If you are not following procedures like these, your projects will suffer problems, quite possibly serious problems. Unfortunately, due to space constraints, we can't discuss every aspect of this basic approach.

Instead, we are concerned here with the build, deployment, and server configuration steps. Notice that the preceding list makes no mention of automation. This is not an accident. Rigorous procedures can be developed without automation and will work fine for projects of a certain size. But for various reasons already discussed, automation can greatly benefit the basic approaches. Many parts of this book focus on using automation.

In this chapter, we will focus on moving an application out of the desktop development environment and into development integration. Because the deployment life cycle starts from the development environment, creating a rigorous process for moving a build out of development can set the tone for a successful deployment. Rigorous process in development can be a model for rigorous deployment in other environments, such as system tests or performance testing. We will focus on these other environments in Chapter 16.

Development and Build Terminology

Before continuing with the remainder of this chapter, we need to define some common terms that we are going to be using. Some of this terminology may be familiar. In order to be consistent with terms, we define several here:

1. ***Runtime or runtime environment:*** A place where code can be executed. Basically, this is a system or set of systems that includes WAS and other needed components where application code is deployed for various purposes.
2. ***IDE:*** An Integrated Development Environment used by an application developer to write, compile, and unit test code. WebSphere Studio is an example of an IDE.
3. ***Unit Test Environment (UTE):*** A runtime where developers run their unit tests. This runtime is often integrated with the IDE to offer developers quick testing and debugging. The WAS unit test environment inside WebSphere Studio is an example of a UTE. If developers do not have access to an IDE with a UTE, it is not uncommon for them to have a WAS install on their desktop for unit testing.
4. ***Verification runtime:*** A runtime needed to verify a successful build. This runtime is usually an application server and usually is located on the same machine where an official build is done. Depending on where the build is done, this runtime can be a UTE inside of an IDE, like WebSphere Studio, or a standalone WAS install on a dedicated build server.
5. ***Code repository:*** A place where developers store versions of code and code history to share with others. Examples of code repositories are Rational Clearcase and the Concurrent Versioning System (CVS). We will use CVS throughout this book because it is readily available. Appendix C, "Setup Instructions for Samples," has information about where you can obtain CVS.
6. ***Golden Workspace:*** A special term we use to represent a desktop snapshot of a deployable version of the code. This is essentially a version of the system loaded into a single IDE instance. This instance represents a final version before an official build.
7. ***Build server:*** A centralized server needed to run automated builds. A centralized build can be used to coordinate projects. Specialized build scheduling and publishing tools, such as Cruise Control or AntHill,1 can be installed on a build server to add an extra level of automation.
8. ***Binary repository:*** A place to store packaged builds that are ready for deployment. Builds must be stored and versioned just like code.

Build and Deployment Models

Now that we have outlined some key concepts, we are going to define three build and deployment models at increasing levels of sophistication. The three models are Assemble Connection Model, Assemble Export Model, and Assemble Server Model. We'll return to the discussion of server configuration later.

ASSEMBLE CONNECTION MODEL

The simplest possible method

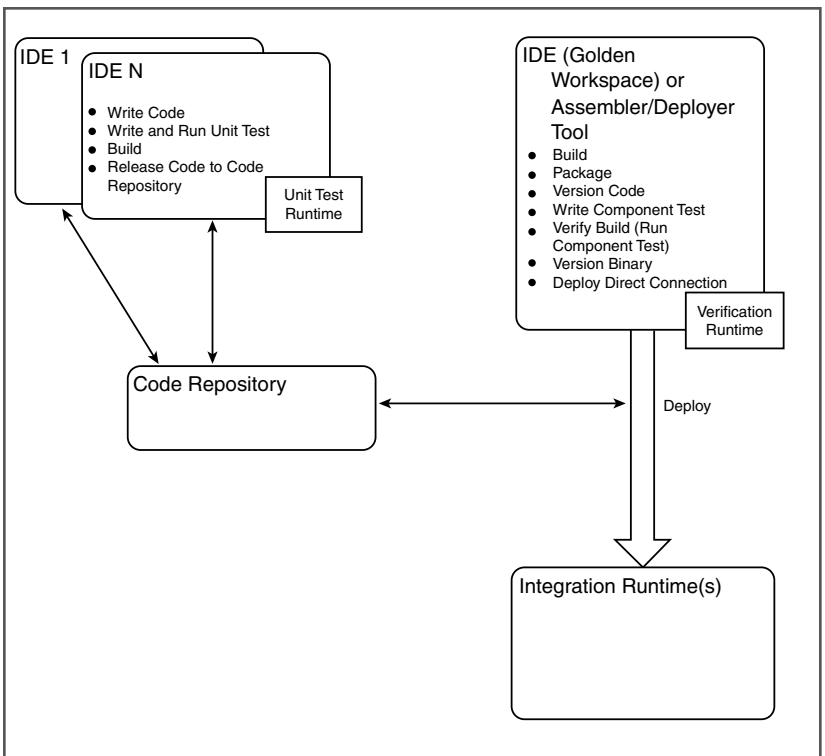


FIG 4-1: ASSEMBLE CONNECTION MODEL

of build and deploy is the Assemble Connection Model. Figure 4-1 illustrates this model.

In this model, build, assembly, and verification steps are all performed from a desktop using either an IDE or an assembly tool. For example, WebSphere Studio contains tools for deploying both server configurations and applications directly from a WebSphere Studio workspace into a runtime environment.

When using this model, it is essential to use a Golden Workspace as part of the official build and deploy process. That is, instead of deploying directly from an uncontrolled developer's desktop, use a dedicated desktop (or at least a dedicated workspace) for creating and exporting to the application server.

This model is simple for small development environments and it is easy to implement. However, it has major drawbacks:

- The development team makes use of deployment shortcuts available to developers. For example, WebSphere Studio has a remote server deployment feature that enables an application to be deployed directly from the development workbench. No EAR is ever produced, and thus the deployment details are hidden from developers. Despite its simplicity, this deployment process is less portable because J2EE defines no way of deploying applications that are not in an EAR or one of the submodules.
- Because no EAR file is produced, there is no versioning of any EAR file. If an environment needs to rollback to a prior version, the build must be reproduced from the source. This can lead to unanticipated errors reproducing the correct build if the source repository tool is not sophisticated or if assemblers do not make use of code repositories correctly.
- It takes great discipline for developers to make this work. It is very difficult to consistently reproduce a build. Deploying directly from the IDE or application source places too many dependencies on idiosyncrasies.
- The process used in development and testing does not usually reflect the process used in production. This means the deployment process for production will mostly likely suffer from a lack of proper testing.
- This process does not scale well. As the artifacts and number of developers grow, the assembler and deployer become a bottleneck. Many artifacts and developers are feeding input to a human that is executing a manual process.

"It is essential to use a Golden Workspace as part of the official build and deploy process."

We strongly oppose this approach. We include it here only as a point of contrast to the next two approaches, which are more reasonable.

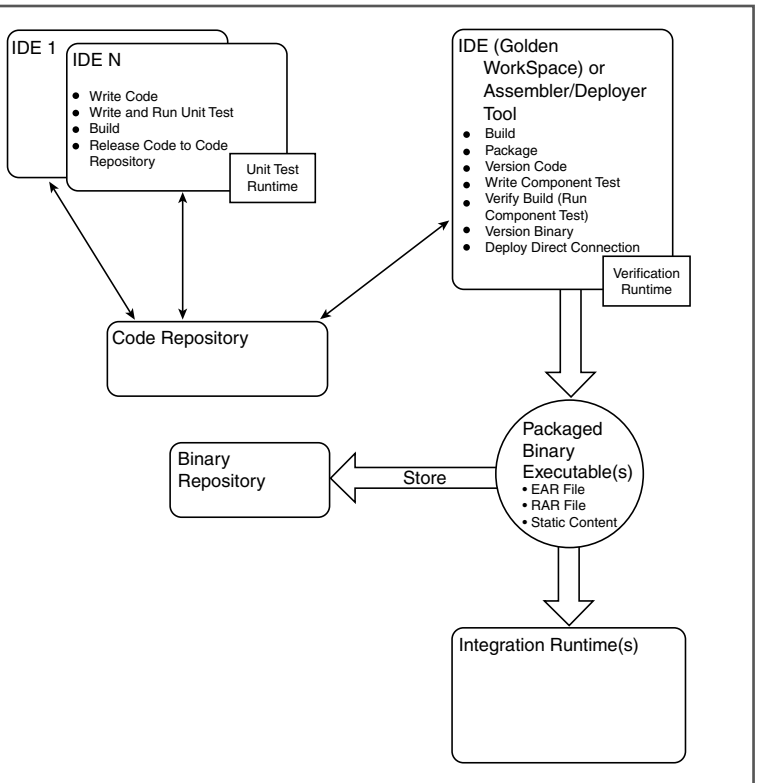


FIG 4-2: ASSEMBLE EXPORT MODEL

ASSEMBLE EXPORT MODEL

The key problem with the previous model is that the deployment step that is used during development does not in any way reflect what is typically done in production. That's unfortunate because this delays the finding of inevitable problems. In the Assemble Export Model, we recognize this weakness and move to a more robust model. Here, the development tools are used to produce the deployable package (either from a Golden Workspace or from deployment scripts). The deployable package is then delivered to the runtime environment and deployed using a well-defined procedure. Figure 4-2 illustrates this model.

This results in a process that is rigorous but potentially time-consuming, depending on the complexity of the application and deployment environment. It is also subject to human error. For systems of reasonable size, this approach will work. With large environments, however, this is often insufficient.

Nonetheless, this is a clear and defined way of producing an EAR file every time. The application is deployed to a runtime similar to how it will be done in production. Therefore, certain errors can be found in this model that could not be found in the previous model. This approach has some nice advantages. Unfortunately, this model still suffers from some drawbacks:

- There is still a scalability issue because someone is needed to create the EAR file from within the IDE (admittedly, this is only a few mouse clicks) and then execute the deployment using the WAS admin tools. In addition, the application assembler has to verify that changes made by various developers work together. As the number of developers increases, this task can become more difficult for one individual.
- The build process is dependent on the development team. In some organizations, this is an issue in itself.

This model has been used successfully in numerous organizations. Furthermore, if the project grows to a larger size, it can be upgraded to the next model.

ASSEMBLE SERVER MODEL

The Assemble Server Model expands on the previous model by externalizing the build process. That is, instead of performing builds from within an IDE or from a developer's workstation, builds are done in a separate environment. Obviously, externalizing builds from the IDE requires automation of the build step.

The need for this model is dependent on some key factors:

- ***The size and complexity of the project:*** Large development efforts often require an externalized build environment to coordinate many pieces.
- ***Build and deploy frequency:*** Projects with demanding schedules often require frequent building and testing in order to meet demanding schedules. There needs to be a way to manage and automate many of the tasks to allow this to happen. By moving builds to dedicated machines, additional opportunities for automation are created.
- ***Multiple development teams:*** Large development efforts often involve multiple development teams, sometimes under different management. In order to manage the different teams, it is best to have a designated environment for builds and one team responsible for it.

Figure 4-3 illustrates the Assemble Server Model. This model solves many of the drawbacks in the other models. However, there are downsides:

- The initial planning and creation of scripts takes significant amounts of time. The scripts need to be tested and verified as much as the application itself.
- The build environment needs to be built and tested. Building environments sometimes takes extra effort that people don't always want to invest. In addition, you must consider that resources may not be available. Having an extra build machine may not be feasible.

Nonetheless, if this approach is combined with automation, there are long-term benefits to using this model, and it will end up saving time for large or complex projects. A server-side automated build is highly scalable because it does not generally depend on the number of artifacts or developers. Of course, substantial effort will need to be invested in the creation of automation scripts. In the next section, we'll talk more about automation.

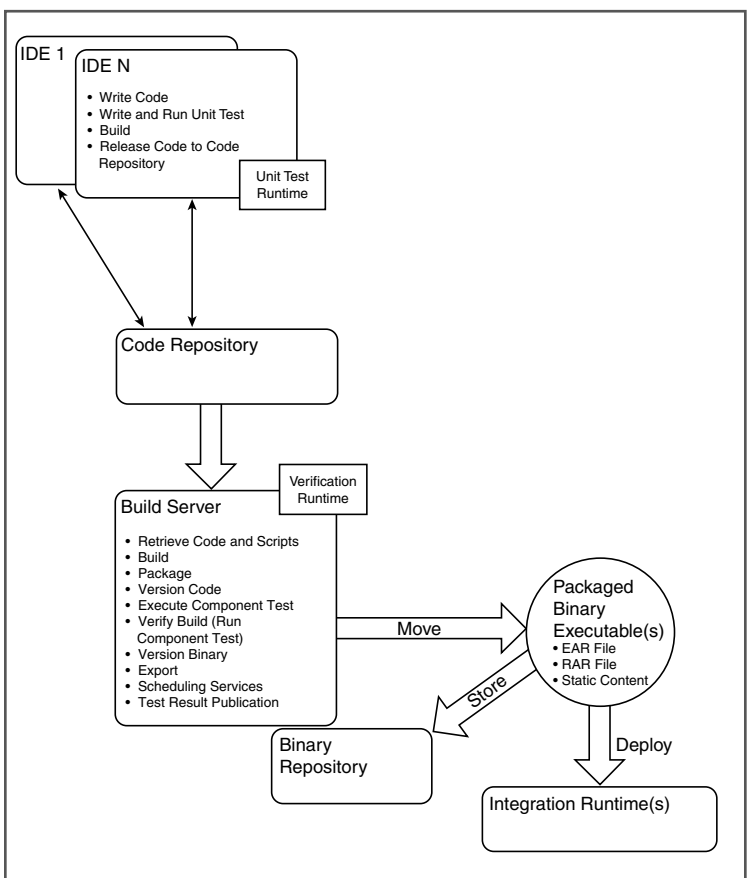


FIG 4-3: ASSEMBLE SERVER MODEL

25 questions that guide the way

Creating a Healthy, Optimized Workflow Environment

BY PRAVEEN K. **CHHANGANI**

As a certified IBM WebSphere MQ Workflow Specialist, Praveen K. Chhangani is part of Prolifics' specialized team of WebSphere consultants that IBM calls upon to service its most challenging customer requirements by providing training, customization, administration and configuration, architecture design, development, and deployment of distributed architectures. Praveen has several years of experience and is well rounded as a developer, analyst, administrator and solutions architect. His extensive experience with IBM WebSphere MQ Workflow, WBI Modeler, Monitor, and Process Choreography has proven invaluable in a full life cycle of projects from requirement analysis, process design and automation, solution design, and development to testing and mentoring for clients such as MCI and Principal Financial Group. pchhangani@prolifics.com

"Workflow" is a term that is typically used within the boundaries of operations and people involved within a given system. Most importantly, today it revolves around the aspects of business integration methodologies by promoting process automation and business level monitoring in real-time while supporting the fluctuations in business growth.

The IBM WebSphere MQ Workflow product provides an innovative integration approach that not only provides solutions enabling increased efficiency via automation involving disparate systems and people, but also leverages the benefits of IBM's industry-leading messaging product, IBM WebSphere MQ. WebSphere MQ provides benefits such as assured delivery of messages, dynamic delivery of workload, and an assurance of comprehensive security choices such as SSL (the Internet standard for secure communication).

When working in a Workflow environment we often see that performance is a critical issue that developers face. The questions below will give you a good diagnosis of areas that you may need to further investigate in order to improve performance.

25 Questions to Ask When Performance Is an Issue

1. HAVE YOU SET UP THE BEST ARCHITECTURE?

To enable your applications to perform and scale to a growing number of users, it is always better, if possible, to use a distributed, three-tier model. Separate the writing of your database logs/files to a different area from where Workflow is installed, either a separate disk on a different file system (if on UNIX), or on a different node – and if the option is available, also install WebSphere on a different node. Depending on how the database is being thrashed and the number of times logs are being updated, these logs tend to grow to enormous sizes. This can cause two problems in particular:

- IO time suffers due to constant thrashing of the database.
- File systems fill up and Workflow goes down because of a lack of disk space.

2. ARE YOU USING CLIENT CONCENTRATOR?

If you have a large number of clients connecting to the default queue manager (usually called FMCQM), it is best to go with a WebSphere MQ setup quite popular in the Workflow world known as the "client concentrator." This separate queue manager serves as a host to the clients and then furnishes the requests to the Workflow queue manager.

3. DID YOU MEASURE THE BASIC WORKFLOW UNIT?

The Basic Workflow Unit (BWU) is a performance capacity planning measurement. A SupportPac is available from IBM that enables you to understand how expensive your source is and perhaps thereafter potentially seek more inexpensive ways to accomplish the same business measures. Here is a link to the SupportPac: www-1.ibm.com/support/docview.wss?rs=203&uid=swg24006573.

4. HAVE YOU EVALUATED THE COST OF YOUR FLOW DEFINITION LANGUAGE (FDL)?

Using BWU calculations you can appraise your FDL costs. You then need to determine if that BWU metric is suitably in line with your hardware.

5. ARE YOU USING BLOCKS OR SUBPROCESSES?

Blocks are less expensive to the Workflow engine and, hence, are recommended over the use of subprocesses. If you need to use subprocesses (pieces of executable designed to be called from other processes), then it may be worth modeling once and using elsewhere as well. Again, keep in mind that the cost for a subprocess is a lot more than that of a block. Use blocks for looping activities and for error handling to a certain extent.

6. DO YOU HAVE MQSERIES TRACING TURNED ON?

Tracing can be used to produce diagnostic messages if you suspect there might be a problem with the code, causing performance to degrade.

7. ARE YOU USING DB2?

Have you worked with your DB2 DBA to optimize and maintain the Workflow databases? As a cleanup process, it is important to run DB2runstats and DB2rebinds on a regular basis (e.g., nightly) to update stats on tables and perform a rebind on execs respectively. The following provides a script to perform DB2 runstat and rebind.

```
db2 select tabname, stats_time from
   syscat.tables where tabschema="FMC"
db2 select pkgname, LAST_ BIND_ TIME from
   syscat.packages where pkgschema=" FMC"
db2 reorgchk update statistics on table all
db2 rebind <pkgname> (for all packages)
```

8. ARE YOU USING ORACLE?

Are you performing the DBMS schema analysis on a regular basis? Be sure to educate the DBA on Workflow applications including how audit trails work in Workflow, situations where table spaces on the Oracle side might need to be increased, etc. Also, as ongoing maintenance Oracle should set up where performance is going down day by day. This may be achieved by running the commands shown below on a regular basis on your Oracle runtime database(s).

- ***execute dbms_utility.analyze_schema('FMC','COMPUTE');***: Where FMC is the database user ID for the Workflow configuration.
  Note: If it takes too long to update the statistics based on all the data for the tables, enter the command below.
- ***execute dbms_utility.analyze_schema('FMC','ESTIMATE',20,");***: Where 20 is the percentage of data used for estimating the statistics.

After updating the statistics, no additional steps are needed to make the database use them. If you are using Oracle, you do not need to worry about the equivalent of db2rebind.

9. WHAT IS THE NUMBER OF WORKFLOW EXECUTION SERVER INSTANCES THAT YOU ARE RUNNING ON YOUR SERVER?

Make sure that number is in compliance with the number of physical CPUs on that box.

10. ARE YOU USING NOTIFICATIONS?

If so, are a lot of notifications being generated?

11. ARE YOU HAVING THE WORKFLOW SCHEDULING SERVER AND RELATED NOTIFICATION CHECK THAT INTERVALS RUN BEYOND THE RECOMMENDED RANGE?

They should not run less than 5 minutes.

12. IF YOU ARE USING MQ WORKFLOW VERSION 3.4, ARE YOU MAKING USE OF THE FMCINTERNALLOOP NOOP (NO OPERATION ACTIVITY) IN YOUR MODELS?

13. DO YOU HAVE STAFF RESOLUTION PERFORMED AT EVERY

STEP...PERHAPS EVEN WHEN IT MAY NOT BE NEEDED?

14. WHAT TYPE OF WEBSPHERE MQ LOGGING IS ACTIVE?

- ***Circular?*** Circular logging rotates the logs every so often and overwrites the existing logs. This is suitable for smaller systems, but linear logging should be taken into consideration for large production systems.
- ***Linear?*** Linear logging will not rotate; it keeps writing to the log files. Be sure to work out your disk space and size for such logs well in advance with your UNIX team in order to allow for enough space. The log sizes can go up very quickly. However, this type of logging provides MQ administrators with more ability to backup/restore queues, which is not much of an option with circular logging.

15. WHAT SYSTEM LOGGING AND MONITORING DO YOU HAVE IN PLACE?

- ***WebSphere MQ Monitoring:*** The more important queues (such as the EXEXMLINPUTQ) and channels should be monitored for being full, or for channels being down or in a state other than desired. Also, there should be monitoring on processes such as the MQ processes running on the system, the listeners, queue managers, channels etc. This level of granularity would provide a very robust architecture for monitoring and error handling/disaster recovery.
- ***Database monitoring:*** It is extremely important to have monitoring capability on the database logs as such logs tend to fill up file systems up pretty quickly. Other things that need to be monitored are the database processes, important database instances, etc.
- ***WebSphere Application Server Monitoring:*** It is important to also maintain a monitoring level on WebSphere Application Server. WebSphere processes, application servers, and related executables must be monitored to allow for quick error notification/recovery.

16. WHAT TYPE OF AUDITING DOES THE SYSTEM PERFORM?

- Are there specific reasons why auditing is turned on? Is it definitely needed?
- If FULL auditing is turned on, are you writing to a specified WebSphere MQ queue (in XML format) or the audit trail database? Although it may seem counter-intuitive, research has shown that auditing to MQ is more expensive than auditing to the database. Auditing to a WebSphere MQ queue is, in my opinion, a little more robust in terms of the overall architecture of the Workflow system, as you can keep track of every single XML message more easily. Scan your FDL for such audits prior to importing and import only if needed.

17. IF INCREASING MEMORY CONSUMPTION IS ALSO AN ISSUE, WHAT IS THE STATUS ON YOUR WORKFLOW RUNTIME THRESHOLD?

By default this is turned off. If volume is extensive, have you turned this feature on in Workflow to take advantage of the recycling of the Workflow execution servers?

18. HAVE YOUR FILE SYSTEM(S) BEEN FLOODED?

19. WHAT ARE THE IO WAIT TIMES ON THE SERVER?

(iostat if on UNIX)

20. WHAT ARE THE CPU TIMES ON THE SERVER?

(vmstat if on UNIX)

21. IF YOU ARE IN A UNIX ENVIRONMENT, ARE YOUR DISKS STRIPPED?

22. WHAT ARE THE ASSIGNED CPU PRIORITIES FOR THE WORKFLOW PROCESSES ON THE SERVER?

(UNIX 'NICE' value)

- Fmcamain
- Fmcemains, etc.

23. HOW MANY WORKFLOW CONFIGURATIONS DO YOU HAVE?

Are all of those configurations having the same performance issues?

24. ARE YOU PASSING LARGE IMAGES THROUGH THE INPUT AND OUTPUT CONTAINERS?

25. ARE YOU USING SUSPEND?

Suspend is a programmatic interface (via APIs) to suspend a process instance. You must have the proper authority to do this. When a process instance is suspended, all work items within the process are also suspended. There is a "processinstance.suspend()" method and a "processinstance.suspenduntil()" method which suspends until a specified DateTime.

Typically, if you can get around the suspend process, it is recommended as it incurs a lot of overhead. An alternate, less expensive option is to use an expiration. When you set expiration on an activity and that expiration duration is hit, then the activity ends (finishes) and the state of the activity becomes "EXPIRED." The process flow then takes the next logical path to another activity (or ends depending on your model). Therefore, you can trick a workflow to use the expiration to create a "wait" activity. For example, if you have a process A ‡ B ‡ C ‡ D ‡ E, and after completing A you would like to wait 30 minutes before starting C, set an expiration of 30 minutes on activity B, and assign B to a "dummy user" who logs on to the Workflow system. The Workflow engine will interpret this as an activity that needs to be performed by the user and will wait for exactly 30 minutes. The "dummy user" will never log in and the flow will then automatically move on to step C after the 30 minutes. Another advantage of expiration is that it can be set within Workflow itself, within buildtime or the modeler products, and set on an activity level; suspend only works on the process level.

Hopefully these 25 tips will guide you on your way to a fully optimized WebSphere MQ Workflow environment. With the power of a streamlined environment, you can align and integrate your organization's resources and capabilities with your business strategies, accelerating the traditionally long-running process flows – thereby cutting overall costs and eliminating errors.

YOUR TESTING SUCKS.

1. Writing test code isn't testing – it's coding.
2. Your developers aren't making testable software.
3. It's impossible to get the business side involved in quality.
4. Your existing testing tools have become shelfware.
5. Your customers have become the primary testers.

LISA from iTKO.

Because everyone should own quality.

iTKO Try a free eval of LISA 2.5 at **www.itko.com**

We understand.

You've probably invested considerable time and money trying to automate your testing. But it just didn't work.

You were sold a tool that let you point and click to test on Day 1, and on Day 2 you were maintaining a test script - inserting bugs into your tests at the same rate as you inserted bugs into the code during development.

The tools were too hard for the QA teams to use, they required intense training and developer knowledge, and the seats were more expensive than a presidential fundraiser.

It's OK. We've seen this scenario at every customer we talk to.

LISA is different.

LISA is a comprehensive, no-code automated testing solution ideal for distributed applications and teams.

Using "inline testing" technology, LISA allows even non-developers to instrument and test web services, J2EE apps, databases and websites.

LISA automates test design, staging and monitoring for unit, regression, load and performance tests. And since LISA is pure Java, she's equally stylish on Linux, OS X, Unix or Win clients.

But don't take this ad's word for it. Try LISA for free and talk to the experts at iTKO. We'll show you how everyone can own quality.

An installation guide

WebSphere Portal Server v5.0.2 on SuSE Linux 8 on zSeries

BY ASIM **SADDAL**

Asim Saddal works in the Portal practice in IBM Software Services for WebSphere.
asaddal@us.ibm.com

This article provides instructions for installing IBM WebSphere Portal Server v5.0.2 on the SuSE LINUX 8 (SLES 8) operating system on IBM zSeries hardware.

Fix pack 2 of WebSphere Portal Server v5.0, also known as WebSphere Portal Server v5.0.2, introduces support for WebSphere Portal on the SuSE SLES for S/390 8.2.4 kernel (SuSE Linux 8) and RedHat for S/390 7.2 platforms. WebSphere Portal Server v5.0.2 runs on WebSphere Application Server v5.0.2. The previous version of WebSphere Application Server v5.0 on Linux on zSeries has been superseded by v5.0.2, which is an entirely new installation. There is no fix pack to update a v5.0 installation to v5.0.2 on Linux on zSeries.

The following IBM Middleware products will be installed:

- WebSphere Application Server v5.0.2
- WebSphere Application Server Enterprise v5.0.2
- WebSphere Portal Server v5.0
- WebSphere Portal Fix Pack 2 (also called WebSphere Portal Server v5.0.2)

Note: This article does not cover the installation process of any databases, Web servers, or LDAP servers. To install these products, refer to the manufacturer's documentation. Also, to perform the configuration of WebSphere Portal Server v5.0.2 with databases, Web servers, or LDAP servers, refer to the WebSphere Portal Info Center.

Prerequisites

Tables 1–3 contain the information you need before beginning the installation of WebSphere Portal Server v5.0.2. Here you will find the operating system requirements for SLES 8, IBM hardware requirements, IBM Middleware component space requirements, and WebSphere installation file requirements.

Hardware Requirements

Table 1 displays the system requirements for SLES 8. Table 2 describes the space requirements for the IBM Middleware components.

Table 3 specifies the files and WebSphere Portal Server CDs that are required for the installation of WebSphere Portal Server v5.0.2.

Install WebSphere Application Server Base v5.0.2

The installation program for WebSphere Application Server Base v5.0.2 is located in the "was/zlinux/linuxs390/WAS50" directory on the WebSphere Application Server Enterprise for zLinux disc, which is included with WebSphere Portal v5.0.2.

1. Start the installation

```
cd /mnt/cd1-14/was/zlinux/
linuxs390/WAS50
./install.sh
```

2. Select a language for the wizard GUI and click Next.
3. The installation wizard opens and a Welcome page appears. Click Next to continue.
4. The license agreement will appear. Click the radio button beside the I accept the terms in the license agreement message if you agree to the license agreement and click Next.
5. Choose Custom install to select installation features. Click Next.
6. Select all features to install except the following and click Next to continue.
   - Embedded Messaging
   - IBM HTTP Server (Optional)
   - Web server plugins (If you choose to install the IBM HTTP Server, then select the appropriate plugin.)
7. Accept the default destination directory. Click Next to continue.
8. If you choose to install the Web server plugin, specify the target directory for selected Web server plugins. Click Next to continue.
9. Accept the default values for the Node Name field and Host Name or IP Address fields. Click Next.
10. Review the summary information and click Next to install the product code or Back to change your specifications.
11. The installation process begins. Depending on your system, this may take up to 10 minutes.
12. When the installation is complete, click Next to register the product, or deselect the check box and click Next to register at a later time.
13. Click Finish to close the installation wizard.
14. The installation is complete.

Verification: WebSphere Application Server Base v5.0.2

Follow these steps to verify the proper installation of the WebSphere Application Server v5.0.2 software.

1. Display the WebSphere Application Server version level by typing the following commands:

```
cd /opt/WebSphere/AppServer/bin
./versionInfo.sh
```

   Review the Version field for the IBM WebSphere Application Server installed product. A value of **5.0.2** indicates a proper installation.
2. Verify the installation of WebSphere Application Server by typing the following commands:

```
cd /opt/WebSphere/AppServer/bin
./ivt.sh
```

   Verify that WebSphere Application Server is running and all verification status values are Passed.
3. The verification is complete.

Install WebSphere Application Server Enterprise v5.0.2

The installation program for WebSphere Application Server Enterprise v5.0.2 is located in the "was/zlinux/linuxs390" directory on the WebSphere Application Server Enterprise for zLinux disc, which is included with WebSphere Portal 5.0.2.

1. Start the installation

```
cd /mnt/cd1-14/was/zlinux/
linuxs390
./install.sh
```

2. Select a language for the wizard GUI and click Next.
3. The installation wizard opens and a welcome page appears. Click Next to continue.
4. The license agreement appears for you to read. Click the radio button beside the I accept the terms in the license agreement message if you agree to the license agreement and click Next to continue.
5. Note that the installation wizard detects a previous installation of WebSphere Application Server V5.0.2. This is an anticipated event in the process; accept the defaults on this panel. Click Next.
6. Choose Custom install to select installation features. Click Next.
7. A panel of selectable installation components displays. Accept the defaults on this panel. Click Next.
8. Review the summary information and click Next to install the product code or Back to change your specifications
9. The installation process begins. Depending on your system, this may take up to 10 minutes.

SYSTEM REQUIREMENTS	DESCRIPTION
Memory	Minimum of 1 GB
DASD	Minimum of three mini-disks • for / • 2 for /opt
RPMs	Packages • compat-2002.8.15-20 • unzip-5.50-57 • vnc-3.3.3r2-172 • pdksh-5.2.14-337

TABLE 1 SLES 8 SYSTEM REQUIREMENTS

10. When the installation is complete, Click Next to register the product, or deselect the check box and click Next to register at a later time.
11. Click Finish to close the installation wizard.
12. The installation is complete.

Verification: WebSphere Application Server Enterprise v5.0.2

Follow these steps to verify the proper installation of WebSphere Application Server Enterprise v5.0.2.

1. Display the WebSphere Application Server version level by typing the following commands.

```
cd /opt/WebSphere/AppServer/bin
./versionInfo.sh
```

   Review the Version field for the IBM WebSphere Application Server Enterprise installed product. A value of 5.0.2 indicates a proper installation.
2. The verification is complete.

Apply WebSphere Application Server v5.0.2 Fixes

WebSphere Portal Server v5.0.2 requires 17 e-fixes to install on the WebSphere Application Server. These are available in the "fixes/<operating_system>/efixes" directory on the WebSphere Application Server Fix Pack disc for your operating system. Or you can download a compressed archive containing the fixes from the WebSphere Portal support site.

Follow these steps to install the WebSphere Application Server v5.0.2 Fixes.

1. Copy the fixes directory and the e-fix installation program to the /tmp/efixes directory.
2. Start the installation program by typing the following commands:

```
cd /tmp/efixes/fixes/linux/
./updateWizard.sh
```

3. Select your preferred language. Click Next.
4. The Welcome to IBM WebSphere Application Server products welcome panel appears. Click Next.
5. Note that the installation wizard detects previous installations of WebSphere Application Server v5.0.2 and WebSphere Application Server Enterprise v5.0.2. This is an anticipated event in the process; accept the defaults on this panel. Click Next.
6. Click the Install fixes radio button. Click Next.
7. Specify the path in the Fix directory field as the following directory and click Next.

```
/tmp/efixes/fixes/zlinux/efixes
```

8. Click the checkboxes to select all fixes. Click Next.
9. Verify the fixes selected for installation. Click Next.
10. A panel displays indicating that the fixes were successfully installed. Click Finish.

COMPONENT	/OPT	/TMP
WebSphere Portal Server	1124 MB	50 MB
WebSphere Application Server	968 MB	245 MB
IBM HTTP Server	30 MB	n/a
Total	2413 MB	295 MB minimum

TABLE 2 IBM MIDDLWARE SPACE REQUIREMENTS

Verification: WebSphere Application Server v5.0.2 Fixes

To verify the proper installation of the WebSphere Application Server v5.0.2 Fixes, display a list of the fixes you installed in the fix pack by typing the following commands:

```
cd /tmp/fixes/linux
./updateSilent.sh –fix –installDir
/opt/WebSphere/Application
```

Note: Although this command is shown on multiple lines for clarity, it should be entered on the same line.

The entries in this list confirm the individual fix packs that you selected for installation.

Install WebSphere Portal Server v5.0

The installation program for WebSphere Portal Server v5.0 is located in the root directory on the Setup CD disc, which is included with WebSphere Portal 5.0.2.

1. Start the installation

```
./installzLinux.sh
```

   Note: You must run the installzLinux.sh command from the directory where the command script is located. Attempting to run the command from another location (for example, /mnt/cdrom/installzLinux.sh) will fail.
2. Select a language for the wizard GUI and click Next.
3. The installation wizard opens and a Welcome page appears. Click Next to continue.
4. The license agreement appears for you to read. Click the radio button beside the I accept the terms in the license agreement message if you agree to the license agreement and click Next to continue.
5. WebSphere Portal Install checks for operating system and software requirements, and then displays a panel of setup options. Choose Custom setup.
6. Click the radio button for Use an existing instance of WebSphere Application Server. Click Next.
7. Assure that the following directory in the Existing WebSphere Application Server Instances field is highlighted. Click Next.

```
/opt/WebSphere/AppServer
```

8. The installer validates the instance of WebSphere Application Server Enterprise install, then specifies an installation directory for WebSphere Portal Server. Accept the default directory. Click Next.
9. Type the Portal administrative username and password. Click Next.

```
Portal administrative user:
wpsadmin
Portal administrative user
password: wpsadmin
Confirm password: wpsadmin
```

10. A panel displays the features to be installed. Click Next.
    - WebSphere Portal
    - WebSphere Portal Content Publishing

FILE	WEBSPHERE PORTAL CDS	DESCRIPTION
C51veml.tar	cd-1-14	WebSphere Application Server V5.0.2 for zSeries
C547DML.taz	cd1-14	WebSphere Application Server Enterprise V5.0.2 for zSeries
C547GML.zip	Fixpack	WebSphere Application Server Fixpack V5.0.2 for zSeries
C5486ML.zip	Setup	WebSphere Portal Server V5 for SLES 8 (Setup)
C54KQML.zip	Fixpack	WebSphere Portal Fix Pack 2, V5.0 (Fix Pack)

TABLE 3 WEBSPHERE PORTAL SERVER V5.0.2 REQUIREMENTS

11. The installation process begins. Depending on your system, this may take up to 45 minutes.
12. When the installation is complete, click Finish.

Verification: WebSphere Portal Server v5.0

Follow these steps to verify the proper installation of the WebSphere Portal Server v5.0 software.

1. Start the WebSphere Portal Server if it is not already running

```
cd /opt/WebSphere/AppServer/bin
./startServer.sh WebSphere_Portal
```

2. Open a Web browser and type in the following URL:

```
http://<hostname>:<port>/wps/portal
```

where hostname is the fully qualified hostname and port is the port where WebSphere Portal is installed; normally it is 9081.

3. The IBM WebSphere Portal Server Welcome page displays. Click Login.
4. The WebSphere Portal Administrative page displays. Note that in the About WebSphere Portal portlet, the version of IBM WebSphere Portal has changed to 5.0.
5. The verification is complete. Click Logout.

"You can download a compressed archive containing the fixes from the WebSphere Portal support site"

Install Fix Pack 2 of WebSphere Portal Server v5.0

The installation program for fix pack 2 of WebSphere Portal Server v5.0 is available in the "fixpack/<operating_system>" directory on the WebSphere Application Server FixPack disc for your operating system. Or, you can download a compressed archive containing the fixes from the WebSphere Portal support site.

Follow these steps to install the WebSphere Application Server v5.0.2 Fixes.

1. Copy the fixes directory and the e-fix installation program to the /tmp/fixpack directory.
2. Set up the Java environment for the update installer by typing the following commands:

```
cd /opt/WebSphere/AppServer/bin
source setupCmdLine.sh
```

3. Ensure that all WebSphere Application Server servers are stopped.

```
./stopServer.sh server1
./stopServer.sh WebSphere_Portal
```

4. Run the WebSphere Portal update installer to apply the fix pack by running the following command:

```
cd /tmp/fixpack
./updatePortal.sh -fixpack
      -installDir "/opt/WebSphere/PortalServer"
      -fixpackDir "/tmp/fixpack/fixpacks"
      -install
      -fixpackID WP_PTF_502
```

Note: Although the command above is shown on multiple lines for clarity, it should be entered on the same line.

After you have installed the fix pack using one of the previous commands, *you must complete the configuration steps in these instructions before proceeding with any other WebSphere Portal configuration*. Failure to complete these steps will result in a partial installation of the fix pack, which can cause unexpected behavior and interfere with the proper functioning of your portal.

5. Perform the configuration steps by typing the following commands:

```
cd /opt/WebSphere/PortalServer/config
./WPSconfig.sh WP-PTF-502 -DPortalAdminPwd=wpsadmin
```

6. The execution of this command will require about 85 minutes. The configuration has completed successfully when you see BUILD SUCCESSFUL at the end.

Verification: Fix-pack 2 of WebSphere Portal Server v5.0

Follow these steps to verify the proper installation of the WebSphere Portal Server v5.0 software.

1. Start the WebSphere Portal Server if it is not already running.

```
cd /opt/WebSphere/AppServer/bin
./startServer.sh WebSphere_Portal
```

2. Open a Web browser and type in the following URL:

```
http://<hostname>:<port>/wps/portal
```

where hostname is the fully qualified hostname and port is the port where WebSphere Portal is installed; normally it is 9081

3. The IBM WebSphere Portal Server welcome page displays. Click Login.
4. The WebSphere Portal Administrative page is displayed. Note that in the About WebSphere Portal portlet, the version of IBM WebSphere Portal has changed to 5.0.2.
5. Click Logout.

A high-level overview of its anatomy and artifacts

Eclipse Modeling Framework

BY PETTER **GRAFF**

Petter Graff is vice president of InferData, Ltd. He has more than 20 years of experience building object-oriented solutions. The last 10 years he has been teaching and consulting on enterprise architectures for Fortune 500 companies worldwide. pgraff@inferdata.com

The Eclipse Modeling Framework (EMF) is an open source code generation tool distributed under the Eclipse umbrella. It is a tool created in the spirit of the OMG's Model Driven Architecture (MDA) and an excellent example of the power of MDA.

EMF is capable of creating sophisticated editors from abstract business models. These editors are implemented as plugins for Eclipse. EMF creates feature complete implementations including persistence, business model implementation, editing framework and editors.

(*Note:* WebSphere Studio supports Eclipse plugins. The Eclipse IDE may also be used to directly develop in WebSphere.)

In this article, we'll look at the anatomy of EMF and what it produces by building a simple Eclipse editor for keeping track of music collections.

EMF and MDA

The process of using EMF is compatible with the MDA approach of OMG, however it is missing some of the essential properties of an MDA tool.

- ***EMF is not a general-purpose code-generation tool.*** It generates Eclipse plugins. It would be difficult to make EMF generate, let's say, a .NET application.
- ***EMF generates code from models.*** Strictly speaking, an MDA tool should generate platform-specific models (PSM) before generating code.

Even though it may not line up with OMG 100%, EMF is one of the most powerful ambassadors for the MDA approach.

Process of Using EMF

In Figure 1 we have outlined the typical process used when developing plugins with EMF.

First, we need to create a business model (OMG's PIM). In EMF, this model is called the *ecore* model. The ecore model contains structural requirements for the implementation of the editor.

Second, we need to configure the code-generation options. These options are stored in a model called the *genmodel*. The genmodel decorates the ecore model with information specific to the solution domain (in our case Java and Eclipse).

The third step is to generate the code. This step uses the ecore model and the genmodel to generate the implementation for the editor plugins.

The generated code can now be tested. It is feature complete, but a rather naive implementation of an editor. In most cases, a developer has to take over the implementation of sections of the generated code. EMF has a feature that allows the developer to flag the code he/she wants to enhance.

After the code has been generated, a developer typically starts an iterative process where various discrepancies are fixed. The discrepancies can be separated into three categories:

- ***Errors in the business model:*** Requires update to the ecore model
- ***Suboptimal configuration of the code generation options:*** Requires changes to the genmodel
- ***Requirements for a more sophisticated implementation than that generated:*** Requires that the developer takes control of sections of the generated code and modifies the code.

EMF has excellent support for this iterative development process, allowing the developers to fix and regenerate.

Creating the Models

Let's illustrate the various stages and artifacts in EMF by creating a simple application to keep track of a music collection.

We first need to create a business model for the information content the editor will manipulate. In EMF, the business requirements are captured in ecore models. The ecore model is really just an XML file with an ".ecore" extension, hence, we could manipulate the XML file directly. However, for convenience EMF lets you define the ecore model in many ways (see Figure 2):

- A built-in primitive editor (this editor is actually an EMF generated editor!)
- Import from XML Schema
- Import from Rational Rose Models
- Import specially annotated Java

Interfaces (actually, this option is a synchronized option. When the Java interfaces change, the ecore model is synchronized and vice versa).

The ecore model for the Music library contains the structural rules for the implementation. These rules can be visualized in a UML class diagram (see Figure 3).

When creating class diagrams, only a few constructs are used:

- ***Classes:*** Represents the object types that we need to persist and manipulate.
- ***Attributes:*** Attributes represents properties persisted on the objects.
- ***Associations:*** Associations represent relationship between objects. The associations have a few different dimensions:
  - *Containment:* Associations may represent containment relationships. The semantic of this in the ecore model is to specify that an object has a lifetime dependency to its container; for example, if an artist is removed, so is all of their work.
  - *Directionality:* Associations may be unidirectional or bidirectional. If an association is unidirectional, it is only possible to navigate the object model from one of the types. E.g., in our example, it is possible to retrieve the works if we know the artist, but it is not possible to find the artist from some work. An association can also be bidirectional allowing navigation in both directions.

It is possible to create the ecore model using the UML notation. There are tools on the market that directly create ecore models (e.g., Omondo-UML). Another alternative is to import Rational Rose models.

The UML diagram shown in Figure 3 defines the ecore model shown in Listing 1. The ecore model can be browsed by the build-in ecore editor shown in Figure 4.

We can now create a genmodel based on the ecore model file. The genmodel adds properties to the various constructs in the ecore file (see Figure 5).

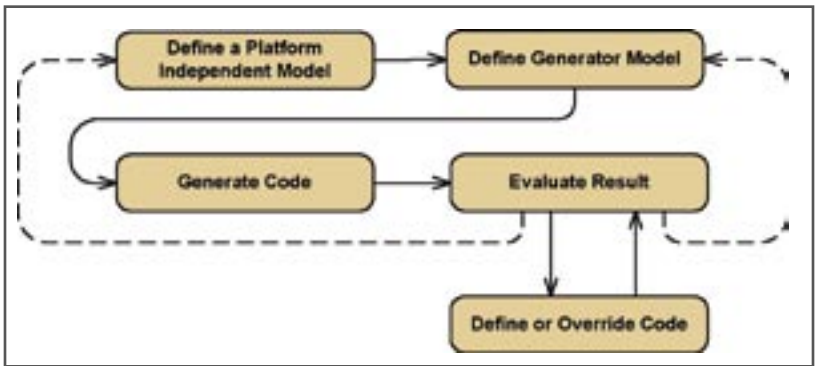


FIG 1: PROCESS OVERVIEW

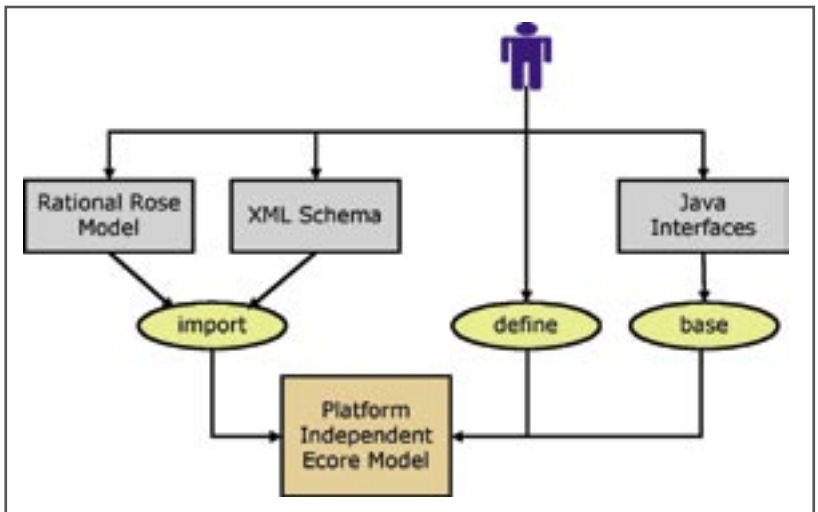


FIG 2: OPTIONS FOR CREATING ECORE MODELS

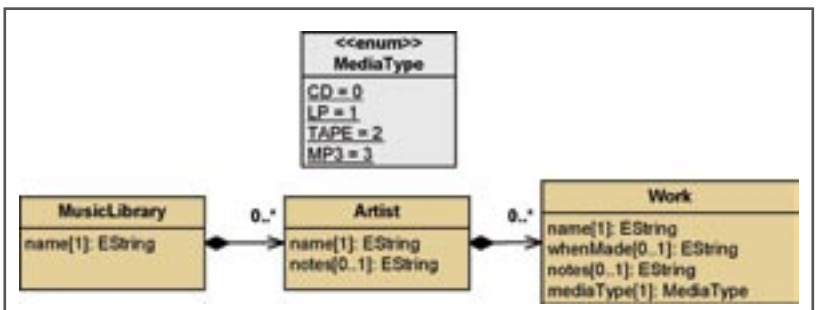


FIG 3: UML MODEL FOR THE MUSIC LIBRARY

The genmodel is also an XML file. The genmodel decorates the ecore model with a set of attributes defining specific configuration for generating Eclipse plugins. It may seem strange to have two models (i.e., why not just add the properties to the ecore model?), but this is essential from the perspective of MDA. If we wanted to create another implementation based on the same business model, we could do so without changing the ecore model.

EMF comes with a point-and-click wizard for creating genmodels from an ecore model, making this task trivial.

Generating Code

Code is generated from the generator model (see Figure 6). The code generator reads the ecore model, the generator models, and a set of code definition templates defined in a template language called JET. The

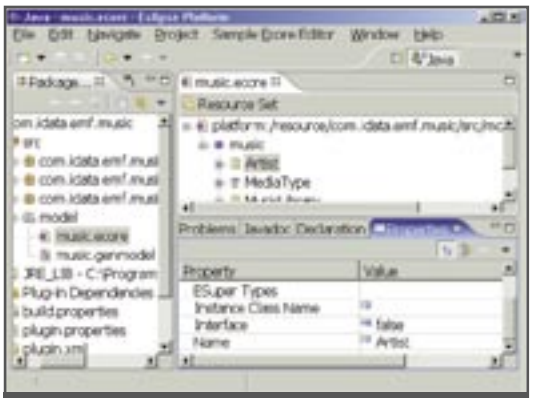


FIG 4: ECORE EDITOR

containing Java implementation files. The framework "expands" the various JET files to generate the Java source files.

What Is Generated?

EMF can create three plugins (see Figure 7):

- ***emf.model:*** The emf.model contains an implementation that is closely aligned with the ecore model. The emf.model plugin contains an object-oriented API that allows programmers to manipulate and persist objects based on the business model.
- ***emf.edit:*** The emf.edit plugin contains user-interface independent editor code. The implementation in this plugin contains adapters that shield the model code from the presentation code. The code also provides a sophisticated command framework with unlimited command stacks to support undo and redo.
- ***emf.editor:*** The emf.editor plugin contains the presentation code for the editor.

The three plugins make up a complete implementation of an editor, allowing users to create and maintain music libraries.

The implementation is based on best practices for developing Eclipse plugins. It uses a set of design patterns to ensure separation and decoupling of concerns. The implementation is highly efficient and its functionality supported far beyond what is typically implemented (e.g., drag-and-drop, unlimited undo/redo, etc).

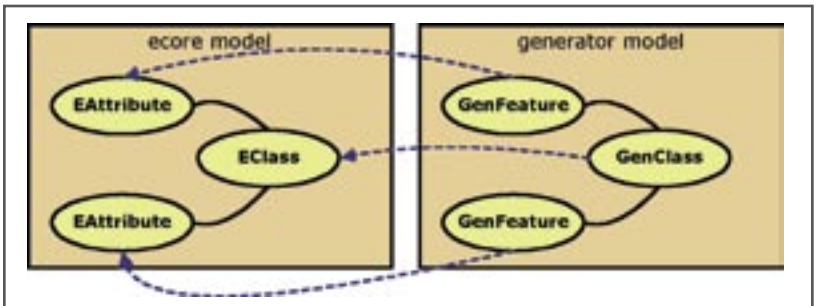


FIG 5: ECORE AND GENMODEL

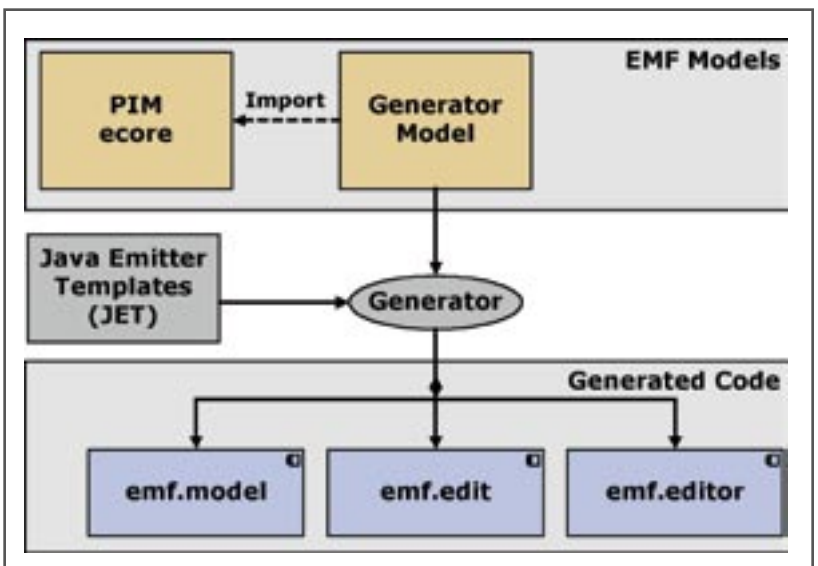


FIG 6: CODE GENERATION FRAMEWORK

JET template is not likely to be modified by most of the EMF users; however, a short description of what they are and the role they play may help to understand the EMF framework.

JET is based on the JavaServer Pages (JSP) syntax (in fact, the JET implementation was based on the Tomcat implementation). A JET template contains a template for files

The Final Result

After generating the code, we can now test or deploy our plugins. The new editor allows us to edit files with a particular extension. The extension name is determined by the code generation options. In our case, we've used the ".music" extension. The new editor recognizes files with a ".music" extension. EMF has also included a wizard for creating new music files (see Figure 8).

Figure 8 shows the running editor. It presents the object structure as a tree. The attribute and associations are modified in a property editor. This is not the only option for presentation. The genmodel allows us to set up various other preferences for how to create the model.

To play around with all the models and the generated plugins, you may download the ecore model, the genmodel and the generated plugins from www.inferdata.com/downloads/emf/emf_intro.

Uses of EMF

The most dominant use of EMF is inside IBM, where it is used for creating editors for their flagship development tool WSAD.

At InferData, we have been using EMF for the following tasks:

- Create persistence implementation for various in-house products
- Create standalone products for the Eclipse platform
- Create quick prototypes to validate complex business models

The use of EMF to create editors is obvious and its benefits are immediate. The last use, to prototype business models, may not be as obvious.

Often, in building large-scale business systems the validation of the business requirements may yield great payback. It is very expensive to find errors late in development. Building formal UML models helps avoid this problem, but there are a limited number of domain experts available that can read the UML models. It has been our experience, that a quick prototype of a business model may widen the understanding of complex business models even though the final target implementation may be on a completely different technology.

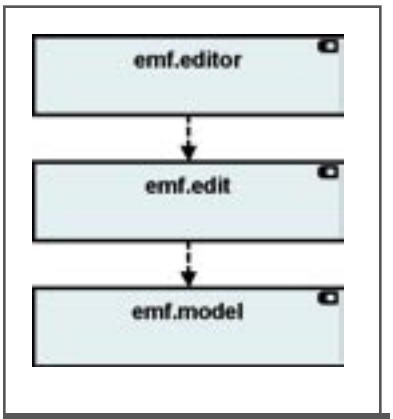


FIG 7: PLUG-IN DEPENDENCY GRAPH

Conclusion

In this article, I've provided a high-level overview of EMF. We've looked at its anatomy and the artifacts it produces.

EMF is a powerful framework for creating Eclipse plugins. Its framework is inspired by the MDA standard emerging from OMG. Even though one may argue that EMF doesn't meet the MDA standard one hundred percent, it is highly practical and provides significant benefit to Eclipse developers.

The main benefits are:

- Significant improvement of productivity.
- High-quality implementation based on best practices for developing Eclipse plugins.
- Excellent separation of concern. Business models remain technology independent; code generation is performed for all that can be code generated and kept separate from the manually developed code.

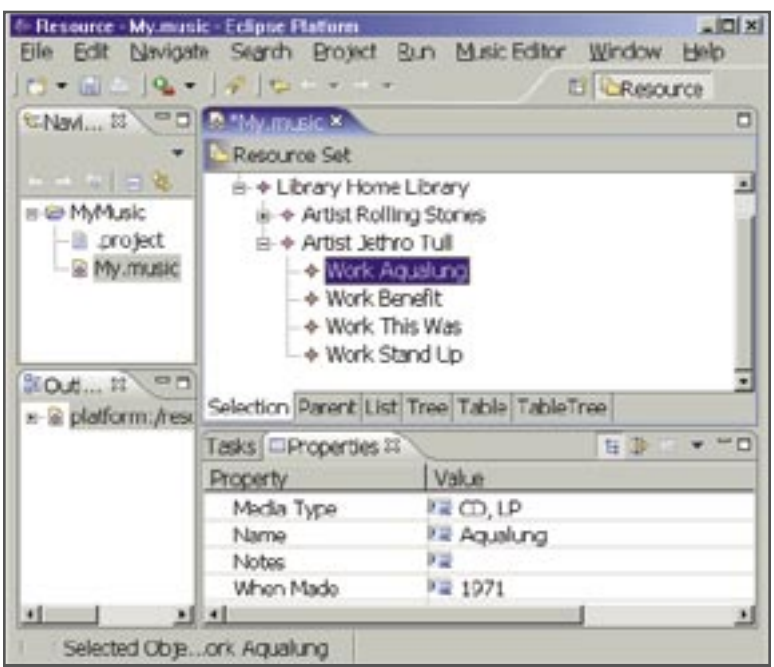


FIG 8: THE MUSIC EDITOR

LISTING 1

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/
Ecore"
  name="music"
  nsURI="http://www.inferdata.com/emf/music"
  nsPrefix="music">
  <eClassifiers
        xsi:type="ecore:EClass"
        name="MusicLibrary">
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="name"
                lowerBound="1"
        eType="ecore:EDataTypehttp://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EReference"
                name="artists" upperBound="-1"
                eType="#//Artist"
containment="true"/>
  </eClassifiers>
  <eClassifiers
        xsi:type="ecore:EClass"
        name="Artist">
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="name" lowerBound="1"
        eType="ecore:EDataType http://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="notes"
        eType="ecore:EDataType http://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EReference"
                name="works" upperBound="-1"
                eType="#//Work"
                containment="true"/>
  </eClassifiers>
  <eClassifiers
        xsi:type="ecore:EClass"
        name="Work">
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="name"
                lowerBound="1"
        eType="ecore:EDataType http://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="whenMade"
                lowerBound="1"
        eType="ecore:EDataType http://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="notes"
        eType="ecore:EDataType http://www.eclipse.
org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures
                xsi:type="ecore:EAttribute"
                name="mediaType"
                lowerBound="1"
                eType="#//MediaType"/>
  </eClassifiers>
  <eClassifiers
        xsi:type="ecore:EEnum"
        name="MediaType">
        <eLiterals name="CD"/>
        <eLiterals name="LP" value="1"/>
        <eLiterals name="TAPE" value="2"/>
        <eLiterals name="MP3" value="3"/>
  </eClassifiers>
</ecore:EPackage>
```

The Importance of Understanding Your Customer's Industry

BY LAUREN STATES

Customer service just isn't what it used to be. I'm not saying it has gone away; most businesses still care about their customers, of course. But it has changed.

Years ago, retailers had fewer customers and knew them personally. They knew what they were inclined to buy because they had been doing business with them for years. Today, retailers and their customers have a huge range of choices in almost every product category, from detergent to athletic shoes. Chain stores and the Internet have made customer relationships less personal while making product preferences more fleeting and the means to purchase more crowded.

Insurance companies could specialize in certain types of coverage, or build a customer set based on a given location. Today, highly volatile industry conditions mean insurers can't rely on a single product or process. They need to look at their businesses from a customer view so they meet their growing needs.

Financial companies are in a business that has always been volatile and high risk. But today they need to sell increasingly complex products with downsized sales staff to an ever-wider range of customers. To serve those customers, they need to get a better handle on the exchange of information in the trading process.

Even governments need to change the way they serve their constituents/customers. With today's safety and security issues, governments need to increase collaboration across agencies and departments so they can securely share the information needed to take quick and appropriate action. Increased cross-government collaboration is also needed to improve day-to-day service, and many constituencies now prefer Web access to government to do things such as apply for permits and pay taxes.

These changes are not inherently bad; they have done wonderful things for competition, efficiency, access, and customer choice. But they mean just about all businesses and organizations have to go about serving their customers differently.

Today, businesses learn their customers' buying patterns and needs by using advanced technologies that can collect data and spot trends. Retailers can tailor advertisements and specials based on customers' individual buying patterns. Insurers and capital market firms can cross-sell and up-sell across product types. Businesses now need to find ways of serving their customers that go beyond knowing when a car lease will expire or offering a low price.

In the coming years, great customer service will require knowing your customer and their industry inside and out. Customer service will mean addressing issues specific to the customer and the industry. Customers will need businesses to solve their problems and meet their needs quickly and efficiently. And businesses will have to do so via an increasing range of channels and locations the customer prefers.

In the software business, this means dedicating deep technical expertise to evaluating, designing, and deploying software solutions geared specifically to solve a customer's business problem. To do this, you have to understand their industry issues as well as their business problems. This combination of strong technical capability and industry knowledge is what will help businesses get the most value from their software and provide the most value to their customers.

Vendors must partner with sales, development, research, and technical support teams to ensure that customer needs are considered from an idea's conception to the successful deployment of a solution. For successful deployment, vendors must have the knowledge to bring in industry expertise, technical testing and processes, basically whatever is needed to address the customers' needs. The vendor becomes the steward for customer success – a trusted advisor – providing a consistent relationship with the client from the initial sale until the solution is up and running and afterward.

Further, to truly serve customers vendors need to help enterprises link their business requirements to their IT environments. This will allow them to integrate their business and IT processes within and beyond their own organizations, allowing them to respond with speed to customers' needs, competitive threats, new opportunities, regulatory requirements and more.

So it's not just about selling the latest feature-rich product. Today businesses need to look at their customers from a holistic perspective, understanding their industry's challenges, helping them address their biggest business problems, and transforming them into businesses that can compete strongly in the market by truly satisfying their customers.

Lauren C. States is a vice president for technical sales and customer deployment in IBM Software Group. She leads the strategy for technical sales support and deployment across IBM's $13 billion software business, and manages its 3,500-member field technical sales support team. Lauren also developed the highly successful management system that assures customer success in deploying IBM middleware throughout global enterprises.

TANGOSOL™

A Cure for Downtime.

Eliminate the single points of failure, add fault tolerance and enable transparent failover for your J2EE applications.

TANGOSOL COHERENCE™ has no single points of failure, ensuring high availability and reliability for your mission-critical J2EE applications. Application state, HTTP sessions and data caches are never at risk. Applications using Coherence keep it up without interruption, even when servers go down.

Try before you buy.
Download a free evaluation license at

www.tangosol.com

Coherence™ is pure Java software that reliably and cost-effectively scales J2EE applications across any number of servers in a cluster. Coherence provides reliable coordinated access to in-memory data that enables safe data caching and user session management for applications in a clustered environment.

Sales +1.617.623.5782 or sales@tangosol.com
Support support@tangosol.com or http://www.tangosol.net
Pricing US$1,995 to US$4,995 per Coherence production CPU
Development licenses are offered free of charge.