SYS-CON

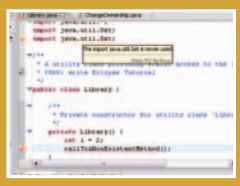
APPLICATION DEVELOPMENT • SOA • TECHNIQUES

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

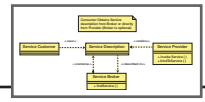
The World's Leading Independent WebSphere Resource

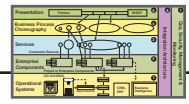
MARCH 2005 VOLUME 4 ISSUE 3

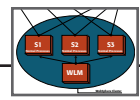


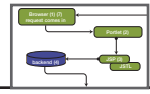
Winning Through Customer Focus

Wily Technology CEO Dick Williams talks with WJ









Deliver better, safer software

in less time, with fewer resources.

Parasoft® Automated Unit Testing & Coding Standards Analysis: specifically designed for under-staffed, over-committed development organizations like yours.

Parasoft products ensure that your entire development organization is meeting requirements and adhering to a common standard. Parasoft products automatically identify security violations, memory errors, runtime exceptions, common defect patterns, and violations of best coding practices.

Increase development effectiveness by allowing:

- **Developers** to correct problems early and reduce time spent chasing bugs
- **Architects** to ensure that development is adhering to a common standard
- **Managers** to improve project visibility, decision making and predictability

The Parasoft product family supports all your major development environments including Java, Web Services, C/C++, .NET and more.

For details go to: www.parasoft.com/Products

Email: Products@parasoft.com Call: 888-305-0041 x3307

PARASOFT®
We make software work.™

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

Open Up!

Portals–Get what you want easily

Ask the Experts

Prolifics is 1 of only 3 WebSphere Service Providers retained by IBM. Open up with portals! With 26 years of industry expertise, Prolifics' certified WebSphere portal experts deliver a personalized, secure, unified view to your enterprise assets.

Our portal solutions build agility into your business — enabling you to reach new markets, achieve customer satisfaction beyond expectation, and ease collaboration within your enterprise. Our teams of specialists work together with your business experts from gathering the business requirements, translating them to a technical specification, to delivering the final production application.

Portal Executive Assessment

Prolifics experts help you articulate the value of portals to your business sponsors and perform a quick business review identifying portal benefits as it applies to your organization.

Portal Workshops and Proof of Concepts

Prolifics demonstrates valuable portal features and translates the breadth of your portal objectives into a step-by-step roadmap to meet your current and future needs.

Portal Jump Start

Prolifics' consultants work with you from the start to install, configure and jump start your team on using WebSphere Portal Software.

Portal Development and Mentoring

Prolifics' innovative approach to building portals drives down your overall cost and transforms the way you conduct business. We mentor and arm your team with the skills to manage, maintain and extend the portal environment.

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

116 John Street, New York, New York 10038
800.675.5419
solutions4websphere@prolifics.com
www.prolifics.com

Prolifics. IBM Business Partner

The SOA of Reality

BY ROGER STRUKHOFF

The Western conception of time relentlessly flying forward is a key underpinning of many societies' continuous focus on improving things, becoming more efficient, getting ever better, and not wasting a minute in the process.

One can guess that this fixation on time's vigilant journey stems from an Age of Science that began several hundred years ago, and is epitomized by Cartesian clocks eternally moving their gears to the tune of a never-ending grind stretching in linear fashion out to near-eternity.

Yet the rotation of those very gears dramatically illustrate another conception, one of continuous cycles, endlessly returning home only to repeat themselves one more time. Whether one thinks of ancient Egyptian flood cycles, lunar calendars, or karmic birth and re-birth, the idea that continuous *cycles* run the universe (and our lives) is at once powerful, easily understood, and valid.

Both conceptions condescend from their ethereal heights to form a rather unbending and brutish reality for IT departments today. You need to design and deploy the most cutting-edge, customer-friendly, Web-centric, fast and efficient, fault-tolerant, real-time applications *right now,* and you had better be thinking in terms of application development cycles and the next iterations and upgrades that will come circling back to torment you before you know it.

IBM WebSphere was conceived to work these problems. Its emphasis on a Services-Oriented Architecture (SOA) is its key enabler—once the SOA light goes on, it becomes easier to understand why you need to take this over-arching, strategic approach to delivering solutions that require a tremendous level of complex, tactical, nitty-gritty development. It's the SOA of Reality, and we're all flying through its cycles!

So, many enterprises have chosen IBM WebSphere to meet the challenges of this Cartesian-karmic existential reality. As the newly named Editor-in-Chief of *WebSphere Journal,* I can report that I am happy to do my part to continue the magazine's leadership as the leading independent editorial voice in the WebSphere community. (My predecessor, Jack Martin, has decided to devote his full energies to his growing consulting practice, and you can continue to reach him at jack@skc.com)

The challenges facing WebSphere developers can be broken into the strategic and the tactical. As we all know, WebSphere is not a *product,* it is a development *environment*. Yet this environment embraces an alphabet soup of IBM and third-party products and technologies that play an integral role within WebSphere.

So focusing on the tactical, *WebSphere Journal* has traditionally excelled in delivering top-notch articles written by your peers on how best to work with these products and technologies, as well as topics related to them. The magazine will continue to do so. For example, this month we have stories about products and topics such as XML, cooperative portlets, the Eclipse UI, Java, and peer recovery.

The magazine has addressed the strategic issues as well, whether through interviews or overarching articles. Our interview tradition continues this issue with our talk with Wily Technology and Barclays Bank. And, we have a strategic article on SOA that sets a high bar for strategic discussion of this most important of issues.

Increasingly known as "soa" rather than "s-o-a," the concept of a Service-Oriented Architecture is what will be driving IT management and programmers mad as an entire generation of application developers flies through too-short time while always cycling back to the same challenges.

It's no secret that a typical big website—whether an Intranet, a B2B supply-chain management system, or one serving consumers directly—is hardly a unified, lean-and-mean fightin' machine, but rather, is actually a welter and concatenation of mainframe legacies, client-server legacies, databases scattered throughout numerous resources and platforms, often with inconsistent user interfaces and functionality levels found throughout.

Those who have chosen to work within IBM WebSphere have made an important first step in addressing these problems and in avoiding similar problems in the future. And we certainly hope that reading *WebSphere Journal* is an important next step.

I can't promise that *WebSphere Journal* can solve all of your Web-centric design and deployment problems. I can't promise that it will ameliorate any of your Cartesian-karmic existential angst. I can say, however, that the magazine will remain an industry leader as we continue to fly—and cycle—through the SOA of reality.

Roger Strukhoff, editor-in-chief of ***WebSphere Journal***, is West Coast Bureau Chief for the SYS-CON News Desk, and President of www.wdva.com. He spent 15 years with Miller Freeman Publications and The International Data Group (IDG), then co-founded CoverOne Media, a custom publishing agency that he sold in 2004. His work has won awards from the American Business Media, Western Press Association, Illinois Press Association, and the Magazine Publishers' Association. Read his blog at http://www.rssblog.linuxworld.com. Contact him at roger@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 ROGER STRUKHOFF
Contributing Editor PATTI MARTIN
Product Review Editor JAY JOHNSON
Editor NANCY VALENTINE
Assistant Editor SETA PAPAZIAN

Writers in This Issue
Ali Arsanjani, Dennis W. Ashby, James Bales, Gary Cernosek, Bill Davis, Richard Gornitsky, Pooja Gupta, Geoffrey Jue, Jack Martin, Stephanie Parkin, Seth Roby, Roger Strukhoff, Douglass Wilson

Subscriptions
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
888-303-5282 201-802-3012
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
CREDITS & ACCOUNTS RECEIVABLE STEVE MICHELIN
ACCOUNTS PAYABLE BETTY WHITE
ADVERTISING DIRECTOR ROBYN FORMA
ADVERTISING MARKETING DIRECTOR DENNIS LEAVEY
ADVERTSING SALES MANAGER MEGAN MUSSA
ADVERTISING SALES MANAGERS KRISTIN KUHNLE, DOROTHY GIL, KIM HUGHES
LEAD DESIGNER LOUIS F. CUFFARI
ART DIRECTOR ALEX BOTERO
ASSOCIATE ART DIRECTORS RICHARD SILVERBERG, TAMI BEATTY, ANDREA BODEN
WEB DESIGNERS STEPHEN KILMURRAY, MATTHEW POLLOTTA
ONLINE EDITOR MARTIN WEZDECKI
CIRCULATION SERVICE COORDINATORS EDNA EARLE RUSSELL, LINDA LIPTON, MONIQUE 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.™

1 888 GET WILY | wilytech.com

wily technology

Energy and Utility Companies Heat Up Their Network Operations

With a newer, smarter approach (from IBM middleware solutions)

BY GEOFFREY JUE & JAMES BALES

Geoff Jue is the World Wide Solutions Offering Executive with IBM's Global Energy and Utilities Industry. He is responsible for IBM's global integrated strategy and offerings for customer operations transformation and driving IBM's On Demand strategy. With over 24 years in the energy industry, Geoff is a frequent speaker on the new generation of outsourcing. Geoff received his MBA from Golden Gate University, Bachelors in Industrial Engineering and Operations Research from UC Berkeley, and a Certificate in Total Quality Management from UC Berkeley.
gjue@us.ibm.com.

With their ubiquitous overhead wires, poles, underground cables and transfer stations, it's easy for energy and utility companies to blend into the landscape and appear to be part of a slow-moving, slow-changing industry. But during the last 10 years, spurred in part by deregulation, many utilities entered new businesses in an effort to increase revenue in their capital-intensive industry. Those forays into areas such as broadband over power lines, telephone service and more brought mixed results while government regulation and customer expectations were increasing.

So today, many utilities are focusing again on their basic assets and customers. With an approximate 10 to 12 percent return on their government-regulated enterprises, most energy and utility companies achieve good business results when costs are kept under control.

Yet new pressures on the industry will make achieving continuing success very challenging. Under pressure from four distinct sources—aging assets and workforce, growing peak demand, the emergence of new technologies, and revenue restraints from regulation and theft—distribution companies around the world are seeking a newer, smarter approach to operating their networks.

Old Networks, New Challenges

Utilities typically operate nearly 50-year-old distribution networks. While a significant portion of this delivery infrastructure is at or near "end-of-life," new microprocessor-based equipment that is replacing traditional electro/mechanical equipment can improve efficiency and responsiveness throughout utility businesses. Energy and utility companies also have a maturing workforce and need to train new workers to manage new systems.

Another challenge to the industry is the growth in peak demand, which is creating the need to augment capacity. Yet regulators are reluctant to approve the recovery of the costs associated with upgrading infrastructure. At the same time, government entities are adding to the pressure to upgrade systems, with fines possible for utilities that don't have adequate reliability records or comply with security standards.

Meanwhile, energy and utility customers are demanding higher levels of service, from more consistent delivery, to faster outage and problem resolution to routine customer service.

Information Technology can help energy and utility companies meet these challenges. Utilities need to integrate their business and IT processes within and beyond their own organizations so they can respond with speed to customer needs, technology issues, market opportunities, regulatory requirements and system problems.

IBM has been working closely with energy and utility entities to forge new partnerships that will help them meet these IT and business challenges. In addition, IBM has 80 percent marketshare for WebSphere MQ in the industry as well as a #1 ranking in 2002 security administration software market worldwide.

Middleware Targets the Industry

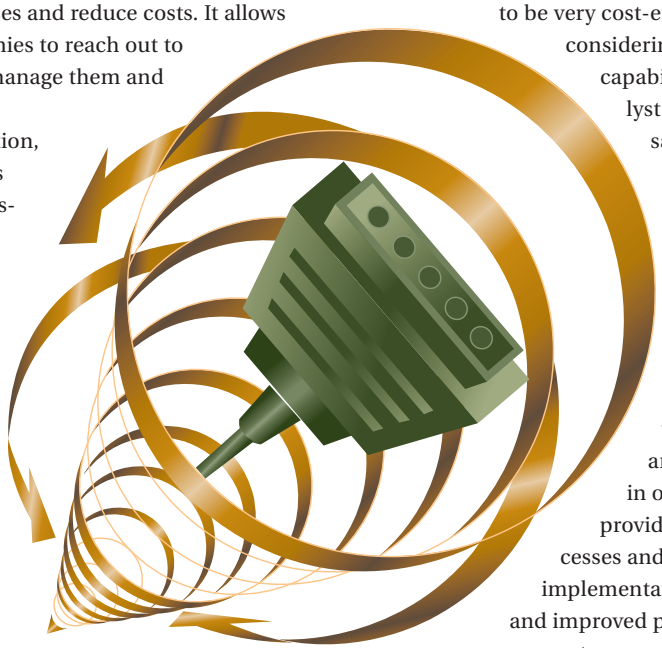
IBM has created middleware solutions designed specifically to address the challenges facing the energy and utilities industry. The solutions are part of IBM's effort to deliver middleware solutions based on customer preference for buying solutions designed for their industry. Each solution draws on the appropriate capabilities from IBM's WebSphere, Lotus, Tivoli, DB2 and Rational middleware brands and IBM's industry-specific middleware, combined with applications from independent software vendors (ISVs)

and industry-expert services.

The solutions are designed to build on existing systems, making better use of what utilities already own. They are designed to reduce costs and to increase the value of information utilities have so they can make better decisions and be more responsive.

The solutions address two key areas: revitalizing utility networks and transforming customer operations, and have been validated with IBM's industry experts, key industry partners and customers. They also contain the security capabilities needed to secure equipment and networks.

The IBM middleware solutions are designed to help utilities harness the vast amount of information they have today—and will have—from their new digitized networks, and use it to significantly improve their effectiveness. The software integrates utility assets with IT and business systems to automate processes and reduce costs. It allows energy and utility companies to reach out to their existing devices to manage them and harvest data.



For network revitalization, the solutions help utilities remotely monitor their distribution network assets, including gas pipelines, electric lines and substations and meters. Utilities can then quickly retrieve key data from them and manage the assets cost-effectively. For example, by connecting these assets to a system of automated data collection, the solution can help utility companies reduce outage time and shorten the customer invoicing process from weeks to minutes.

The solutions also help improve the management and productivity of the mobile workforce by providing anywhere, any time access to the utility's systems. This improves the timeliness and accuracy of data.

Lots of Data, Little Information

Currently, many utilities have lots of data but little useful information. Utilities must integrate their data with business systems to make it useful. When they do, they are able to connect people, processes and information so they can rapidly respond to changing conditions and business requirements.

Advantages to doing so include the ability to use real-time data to improve network reliability, security, and performance, including faster outage detection, management and communication. In addition, providing mobile workers with access to the network improves their productivity and data accuracy across the business. As a result, management of assets is streamlined and made more cost-effective.

Gaining access to on-the-fly data also helps energy and utility companies more efficiently meet regulatory requirements, including Sarbanes-Oxley and FERC Urgent Action 1200 compliance as well as the Department of Transportation pipeline integrity act.

The integration of back-end systems with real-time updates from field crews and remote assets is also essential for energy and utility companies to improve their customer operations. They need to do so because customers are demanding improved service, including self-service and more effective contact centers, which utilities have been unable to deliver across all channels. One of the reasons is that many utilities have multiple customer information systems, due to acquisitions or customer segmentation, which has made efforts to improve operations and cut costs very difficult.

Any improvements to their customer operations have to be very cost-effective, so many utilities are considering implementing self-service capabilities, which IT industry analyst Forrester Research estimates save 90 percent on each transaction compared to service from a customer service representative.

Be SAFE

IBM's middleware solutions are designed to extend and upgrade current customer information systems and leverage the information in other company systems. They provide tools to develop new processes and functions that allow for the implementation of self-service capabilities and improved productivity and utilization among customer service representatives.

The result is improved service to customers, which helps reduce the churn of customers among providers. It also results in reduced costs and streamlined system management for the utility.

In addition to the solutions, IBM has developed a way for utilities to strategically organize, build upon and operate their IT assets for maximum efficiency and flexibility - a framework design known as SAFE: Solution Architecture For Energy & Utilities. This is a flexible architectural framework utilizing open standards and interchangeable components which helps companies leverage and extend existing IT investments, which can result in easier migrations, deferred capital investments and reduced operational costs.

IBM's solutions work with the energy industry's most widely used applications from vendors and they are built using open-standards. The middleware can help utility customers move to automated services, system monitoring and management while helping reduce costs and integrating their data. This data integration—to obtain information at the right time, to the right organization—is critical for utilities to be competitive in the future.

Jim Bales is the Global Offering Executive for Utility Network Revitalization (UNR) Solutions in the Energy and Utilities Market. where he has global responsibility for leading the development of integrated solutions, strategy, and identification and management of key partner relationship for T&D solutions. With 20 years of electric and gas utility industry and consulting experience, Jim has considerable background and experience in the areas of business strategy, IT system implementation, and operational transformation related to T&D operations. Jim has a Bachlors degree in Business for Eastern Illinois University and a MBA from Northen Illinois University. Prior to joining IBM, Jim worked for"Big5"and top tier management consulting firms and held various management positions in the power generation and delivery business units at a major electric utility.
jgbales@us.ibm.com

How to Identify, Specify, and Realize Services for Your SOA

SOA requires a strategy. It is not a product.

BY ALI ARSANJANI

Dr. Ali Arsanjani is a senior technical staff member and chief architect of the SOA and Web Service Center of Excellence in IBM global Services. He has 21 years of experience in software development and architecture. He holds a PhD in computer science from DeMontefort University. Ali's areas of expertise include patterns, component-based and service-oriented software architecture, and methods. arsanjan@us.ibm.com

There has been a lot of buzz and hype, some factual, some not so well-founded, surrounding the opportunities presented by Service-oriented Architectures (SOA) and their implementation as Web services. Analysts have predicted, pundits have professed, professors have lectured. And companies have scurried to sell what they had as SOA products, often missing the point that SOA is not a product.

SOA is about bridging the gap between business and IT through a set of business-aligned IT services using a set of design principles, patterns, and techniques. This article discusses the highlights of service-oriented modeling and architecture, and the key activities needed for the analysis and design required to build a Service-Oriented Architecture (SOA). It stresses the importance of addressing the techniques required for the identification, specification and realization of services, their flows and composition, as well as the enterprise-scale components needed to realize and ensure the quality of services required of an SOA.

A huge demand exists for the development and implementation of SOAs. Gartner has predicted that by 2008, for example, more than 60 percent of enterprises will use SOA as a "guiding principle" when creating mission-critical applications and processes. But if SOA is not just about the products and standards that help realize it—through Web services, for example—then what additional elements are needed to realize an SOA?

Start with Modeling

Service-oriented modeling requires additional activities and artifacts that are not found in traditional object-oriented analysis and design (OOAD). Additional important considerations exist that need to be taken into account. For one thing, current OOAD methods do not address the three key elements of an SOA: *services*, *flows*, and *components* realizing services. You also must be able to explicitly address the techniques and processes required for the identification, specification and realization of services, their flows and composition, as well as the enterprise-scale components needed to realize and ensure the quality of services required.

Second, a paradigm shift needs to occur in order to appreciate the distinct requirements of the two key roles in an SOA: the service provider and service consumer. Third, applications assumed to be built for one enterprise or line of business must now aspire to be used in a supply chain and be exposed to business partners who might compose, combine, and encapsulate them into new applications. This is the notion of the service ecosystem.

This is a slight step up from just "distributed objects." SOAs need to be about the value created through the network effect. An example would be when parties leverage a combination of Amazon.com with Google's search services and combine them with eBay services to build their own hybrid solutions. Or when a travel agency reaches deep into an airline reservation system and coordinates it with a rental car agency and hotel chain, updating their records and sending the itinerary to your electronic organizer.

It's also critical to understand what you are aiming to produce: *what is an SOA, and what does it look like?* After defining the notion and concepts behind an SOA, you need to understand the layers of an SOA. You also need to know to record key architectural decisions about each layer that help you in building an SOA blueprint for your

project, line of business, enterprise-wide effort, or value-chain that you are trying to integrate, and come up with a set of services, flows, and components that implement the SOA.

Whatever the application, you need much more than just good tools and standards to successfully create an SOA. You need some prescriptive steps to support your SOA life cycle; techniques for the analysis, design, and realization of services, flows, and components. Therefore, for anyone interested in enterprise application development, it's crucial to understand the detailed steps involved in service-oriented modeling and architecture.

Conceptual SOA Model

A valid SOA conceptual model is based on an architectural style that defines an interaction model between three primary parties:

- The service provider, who publishes a service description and provides the implementation for the service
- A service consumer, who can either use the uniform resource identifier (URI) for the service description directly, or can find the service description in a service registry and bind and invoke the service.
- The service broker, to provide and maintain the service registry (although nowadays public registries are not in vogue.)

A meta-model showing these relationships is depicted in Figure 1.

An SOA is an enterprise-scale IT architecture for linking resources on demand. In an SOA, resources are made available to participants in a value net, enterprise, line of business (typically spanning multiple applications within an enterprise or across multiple enterprises). It consists of a set of *loosely-coupled, business-aligned* IT services that collectively fulfill an organization's business processes and goals. The architecture style defining an SOA describes a set of patterns and guidelines for services that, because of the separation of concerns between description, implementation, and binding, provide unprecedented flexibility in responsiveness to new business threats and opportunities.

A service is a software resource (discoverable) with an externalized service description. This service description is available for searching, binding, and invocation by a service consumer. The service provider realizes the service description implementation and also delivers the quality of service requirements to the service consumer. Services should ideally be governed by declarative policies and thus support a dynamically re-configurable architectural style.

You can choreograph these services into composite applications and invoke them through standard protocols, as shown in Figure 2.

Business agility is gained by IT systems that are flexible, primarily by separation of interface, implementation, and binding (protocols) offered by an SOA, allowing the deferral of the choice of which service provider to opt for at a given point in time based on new business requirements, functional and non-functional (for example, performance, security, and scalability.)

You can reuse the services across internal business units or across the value chains among business partners in a fractal realization pattern. Fractal realization refers to the ability of an architectural style to apply its patterns and the roles associated with the participants in its interaction model in a composite manner. You can apply it to one tier in an architecture and to multiple tiers across the enterprise architecture. Among projects, it can be between business units and business partners within a value chain in a uniform and conceptually scalable manner.

Another facet that must be addressed is service-oriented modeling—a service-oriented analysis and design (SOAD) process for modeling, analyzing, designing, and producing an SOA that aligns with business analysis,

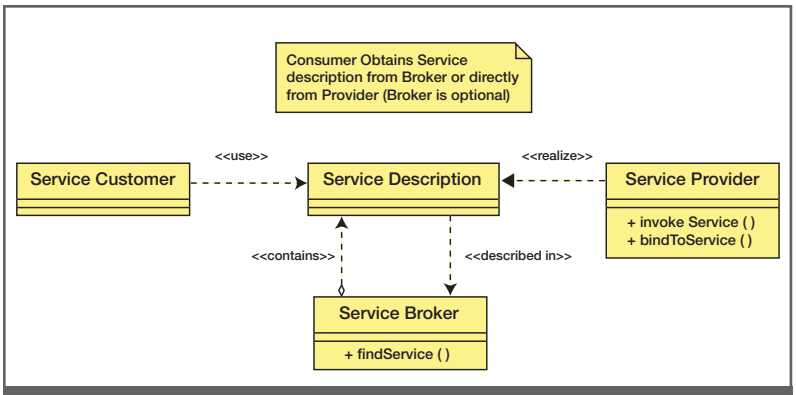


FIG 1: CONCEPTUAL MODEL OF A SOA ARCHITECTURAL STYLE

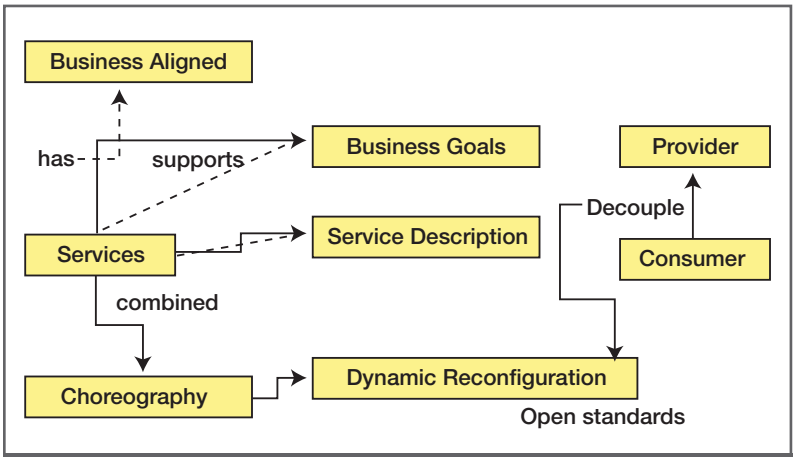


FIG 2: ATTRIBUTES OF AN SOA

1. **Scope** <what area of the enterprise is this architecture for?>
2. **Operational systems layer**
   1. Packaged applications
   2. Custom applications
   3. Architectural decisions
3. **Enterprise components layer**
   1. Functional areas supported by this enterprise components
   2. <What business domains, goals and processes are supported by this enterprise components>
   3. Decisions regarding governance
      1. <Criteria by which something is elected as an enterprise components within this client organization>
   4. Architectural decisions
4. **Services layer**
   1. Categorized portfolio of services
   2. Architectural decisions
5. **Business process and composition layer**
   1. Business processes to be represented as choreographies
   2. Architectural decisions
      1. <Which processes need to be soft-wired into choreographies and which will be built into applications?>
6. **Access or presentation layer**
   1. <Document implications of Web services and SOA on this layer; if any. For example, use of portlets that invoke Web services at the user interface level and the implications on the functioning of that layer>
7. **Integration layer**
   1. <Include considerations of an ESB>
   2. <How are we going to ensure the service-level agreements (SLAs) and quality of service (QoS) required by clients of the services provided?>
   3. Security issues and decisions
   4. Performance issues and decisions
   5. Technology and standards limitations and decisions
   6. Monitoring and management of services
      1. Description and decisions

TABLE 1: SOA ARCHITECTUREDOCUMENT TEMPLATE

processes, and goals. You first need to look at what you intend to build: namely an SOA and its layers. Then you can address the main activities and techniques needed to create an SOA.

Let's assume that you need to migrate a portion of a banking line of business that has a self-service accounting system to an SOA. In order to migrate to the SOA, you need some additional elements that go beyond service modeling. These include:

- **Adoption and maturity models**. Where is your enterprise at in the relative scale of maturity in the adoption of SOA and Web Services? Every different level of adoption has its own unique needs.
- **Assessments.** Do you have some pilots? Have you dabbled into Web services? How good is your resulting architecture? Should you keep going in the same direction? Will this scale to an enterprise SOA? Have you considered everything you need to consider?
- **Strategy and planning activities.** How do you plan to migrate to an SOA? What are the steps, tools, methods, technologies, standards, and training you need to take into account? What is your roadmap and vision, and how do you get there? What's the plan?
- **Governance.** Should existing API or capability become a service? If not, which ones are eligible? Every service should be created with the intent to bring value to the business in some way. How do you manage this process without getting in the way?
- **Implementation of best-practices.** What are some tried and tested ways of implementing security, ensuring performance, compliance with standards for interoperability, designing for change?

In addition to identification, specification, and realization, the service-oriented modeling approach includes the techniques required for deployment, monitoring, management, and governance required to support the full SOA life cycle.

So the above discussions on migration to SOA and the additional activities after realization deserve an article of their own, which I will get to in a subsequent column in this series. For now, let's assume that you scoped the project, and a focal point for transformation of existing systems or services to a new set of systems and services has been defined. You can now start service-oriented modeling to build your service-oriented architecture.

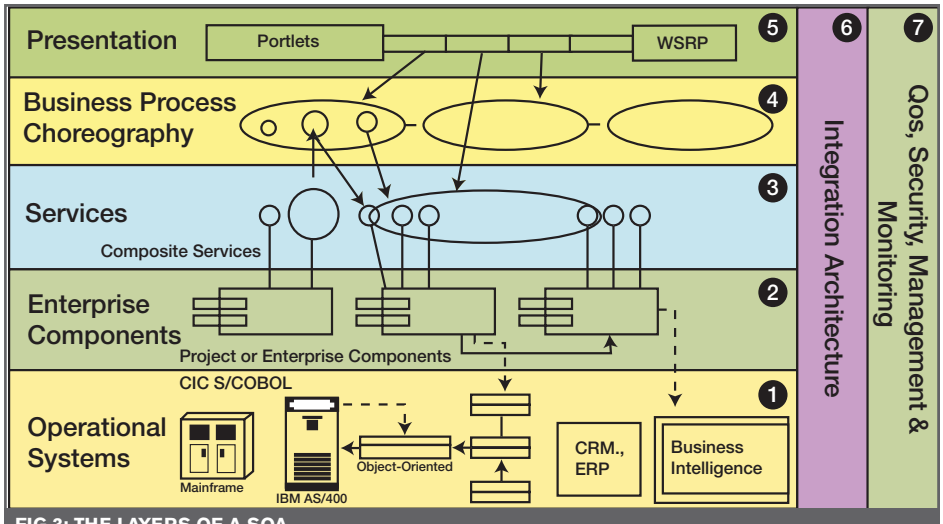


FIG 3: THE LAYERS OF A SOA

An Architectural Template

An abstract view of SOA depicts it as a partially-layered architecture of composite services that align with business processes. Figure 3 depicts a representation of this type of architecture. The relationship between services and components is that enterprise-scale components (large-grained enterprise or business line components) realize


Middleware is Everywhere.
Can you see it?
1
2
3
4
5
IBM
WebSphere.
Key
1. Deductible viewed quickly.
2. Claim info filed accurately.
3. Vendor receives complete data.
4. Quotes researched easily.
5. Great service boosts policy sales.
MIDDLEWARE IS IBM SOFTWARE. Powerful WebSphere software. For an integrated IT environment that makes your business more flexible. WebSphere delivers a secure, scalable platform. Enabling a service-oriented architecture. So your business can respond faster. More efficiently. To partners, vendors and customers. With no ripping. No replacing. No headaches. No kidding.
Middleware for the on demand world. Learn more at ibm.com/middleware/flexible
ON DEMAND BUSINESS
IBM, the IBM logo, WebSphere and the On Demand logo are registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries. ©2005 IBM Corporation. All rights reserved.

Role	Activities in this role				
Consumer View	Service identification	Service categorization	Service exposure decisions	Choreography of composition	Quality of service
Provide View	Component identification	Component specification	Service realization	Service management	Standards implementation
	Service allocations to components	Layering the SOA	Technical prototyping	Product solution	Architectural decisions (state, flow, dependencies)

FIG 4: ACTIVITIES OF SERVICE-ORIENTED MODELING

the services and are responsible for providing their functionality and maintaining their quality of service. Business process flows can be supported by a choreography of these exposed services into composite applications. An integration architecture supports the routing, mediation, and translation of these services, components, and flows using an Enterprise Service Bus (ESB). The deployed services must be monitored and managed for quality of service and adherence to non-functional requirements.

For each of these layers, you must make design and architectural decisions. Therefore, to help document your SOA, you might want to create a document consisting of sections that correspond to each of the layers. Table 1 contains a template for your SOA architecture document.

SOA Layer Descriptions

Now, let's describe each layer in greater detail and discuss the composition of each of these layers.

Layer 1: Operational systems layer. This consists of existing custom built applications, otherwise called *legacy* systems, including existing CRM and ERP packaged applications, and older object-oriented system implementations, as well as business intelligence applications. The composite layered architecture of an SOA can leverage existing systems and integrate them using service-oriented integration techniques.

Layer 2: Enterprise components layer. This is the layer of enterprise components that are responsible for realizing functionality and maintaining the QoS of the exposed services. These special components are a managed, governed set of enterprise assets that are funded at the enterprise or the business unit level. As enterprise-scale assets, they are responsible for ensuring conformance to SLAs through the application of architectural best practices. This layer typically uses container-based technologies such as application servers to implement the components, workload management, high-availability, and load balancing.

Layer 3: Services layer. The services the business chooses to fund and expose reside in this layer. They can be discovered or be statically bound and then invoked, or possibly, choreographed into a composite service. This service exposure layer also provides for the mechanism to take enterprise scale components, business unit specific components, and in some cases, project-specific components, and externalizes a subset of their interfaces in the form of service descriptions. Thus, the enterprise components provide service realization at runtime using the functionality provided by their interfaces. The interfaces get exported out as service descriptions in this layer, where they are exposed for use. They can exist in isolation or as a composite service.

Level 4: Business process composition or choreography layer. Compositions and choreographies of services exposed in Layer 3 are defined in this layer. Services are bundled into a flow through orchestration or choreography, and thus act together as a single application. These applications support specific use cases and business processes. Here, visual flow composition tools such as IBM WebSphere® Business Integration Modeler or Websphere Application Developer Integration Edition can be used for the design of application flow.

Layer 5: Access or presentation layer. Although this layer is usually out of scope for discussions around an SOA, it is gradually becoming more relevant. I depict it here because there is an increasing convergence of standards, such as Web Services for Remote Portlets Version 2.0 and other technologies, that seek to leverage Web services at the application interface or presentation level. You can think of it as a future layer that you need to take into account for future solutions. It is also important to note that SOA decouples the user interface from the components, and that you ultimately need to provide an end-to-end solution from an access channel to a service or composition of services.

Level 6: Integration (ESB). This layer enables the inte-

gration of services through the introduction of a reliable set of capabilities, such as intelligent routing, protocol mediation, and other transformation mechanisms, often described as the ESB. Web Services Description Language (WSDL) specifies a binding, which implies a location where the service is provided. On the other hand, an ESB provides a location independent mechanism for integration.

Level 7: QoS. This layer provides the capabilities required to monitor, manage, and maintain QoS such as security, performance, and availability. This is a background process through sense-and-respond mechanisms and tools that monitor the health of SOA applications, including the all-important standards implementations of WS-Management and other relevant protocols and standards that implement quality of service for an SOA.

Combining Top-Down with Bottom-Up

When developing an SOA, you also need to be able to combine a top-down, business-driven approach with a bottom-up approach, leveraging legacy investments. The service-oriented modeling approach provides modeling, analysis, design techniques, and activities to define the foundations of an SOA. It helps by defining the elements in each of the SOA layers and making critical architectural decisions at each level. It does so using a combination of a top-down, business-driven manner of service identification coupled with a stream of work that conducts service identification through leveraging legacy assets and systems.

In this way, high-level business process functionality is externalized for large-grained services. Smaller-grained services—those that help realize the higher level of services—are identified by examining the existing legacy functionality and deciding how to create adaptors and wrappers, or componentizing the legacy systems to externalize the desired functionality often locked within the system. Finally, using goal-service modeling, you use a cross-sectional approach to cut down the sheer number of candidate services that might already be identified.

A more judicious approach would be to first do top-down, then goal-service modeling, and finally bottom-up legacy analysis of existing assets. The message is: the faster you scope the project down to a manageable and realistic set, the sooner you can realize value by focusing on key services to expose with service descriptions that form the cornerstone of the SOA.

This combination of functional business aspirations and leveraging of existing investments in legacy systems provide a potent solution to organizations that want to have quick wins and migrate their enterprise to a modern SOA. Consolidation of software applications through service-oriented integration thus becomes possible.

Service-oriented integration is an evolution of

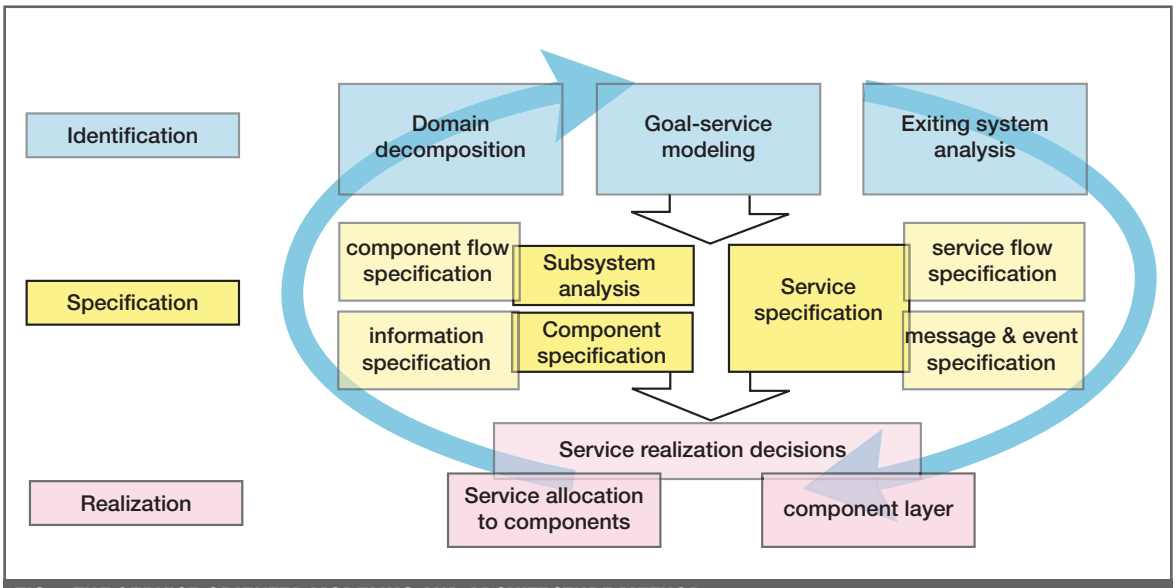


FIG 5: THE SERVICE-ORIENTED MODELING AND ARCHITECTURE METHOD

Enterprise Application Integration (EAI) in which proprietary connections are replaced with standards-based connections over an ESB notion that is location transparent and provides a flexible set of routing, mediation, and transformation capabilities.

Analysis and Design of Services

Unlike with your comfortable world of objects, you need to take into account two perspectives in an SOA, that of the service consumer and service provider. (The service broker is currently not mainstream and will be covered in a later venue.)

The design strategy for an SOA does not start from the bottom-up as is often the case with a Web services-based approach. You must remember that SOA is more strategic and business-aligned. Web services are a tactical implementation of SOA. A number of important activities and decisions exist that influence not just integration architecture but enterprise and application architectures as well. They include the activities from the two key views of the consumer and provider described in Figure 4. This figure shows the activities that are typically conducted by each of the roles of provider and consumer. Note that the provider's activities are a superset of the consumer's activities. For example, the provider would also be concerned with service identification, categorization, etc.

In many cases, the differentiation of the roles comes from the fact that the consumers specify the services they want, often search for it, and once they are convinced of the match between the specification of the service they are looking for, and that provided by a service provider, they bind and invoke the service as needed.

The provider, in turn, needs to publish the services they are willing to support; both in terms of functionality and most importantly in terms of the QoS that consumers will require. This implicit contract between consumer and provider might mature into an explicit contract in terms of SLAs, negotiated either electronically or through business and legal venues.

The activities described above can be depicted to flow within the service-oriented modeling and architecture method, as shown in Figure 5.

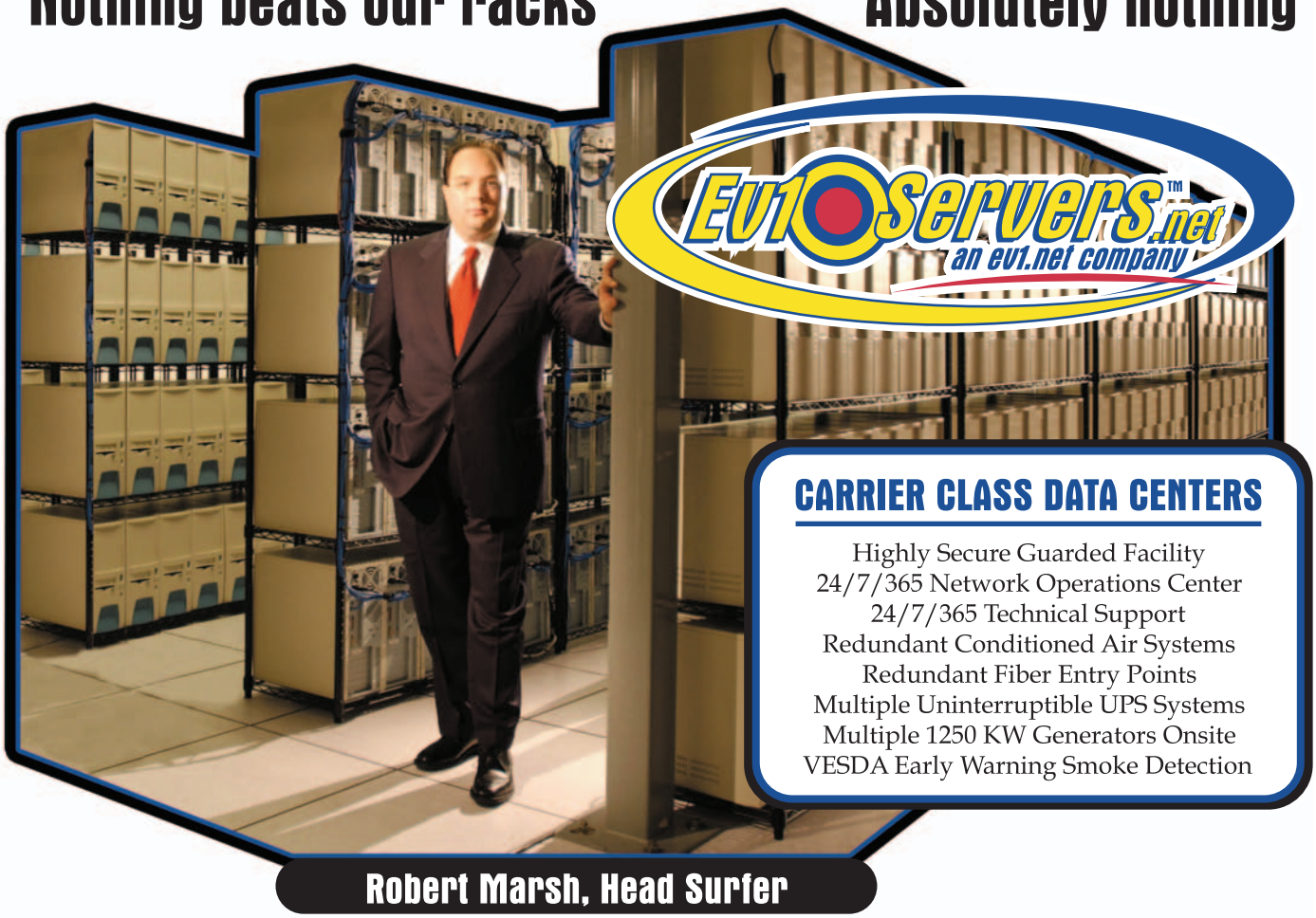
The process of service-oriented modeling and architecture consists of three general steps: identification, specification and realization of services, components and flows (typically, choreography of services).

Service identification. This process consists of a combination of top-down, bottom-up, and middle-out techniques of domain decomposition, existing asset analysis, and goal-service modeling. In the top-down view, a blueprint of business use cases provides the specification for business services. This top-down process is often referred to as *domain decomposition*, which consists of the decomposition of the business domain into its functional areas and subsystems,

"When developing an SOA, you also need to be able to combine a top-down, business-driven approach with a bottom-up approach, leveraging legacy investments"

Nothing beats our racks

Absolutely nothing



CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299*** *Instant Activation!*

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

including its flow or process decomposition into processes, sub-processes, and high-level business use cases. These use cases often are very good candidates for business services exposed at the edge of the enterprise, or for those used within the boundaries of the enterprise across lines of business.

In the bottom-up portion of the process or existing system analysis, existing systems are analyzed and selected as viable candidates for providing lower cost solutions to the implementation of underlying service functionality that supports the business process. In this process, you analyze and leverage API's, transactions, and modules from legacy and packaged applications. In some cases, componentization of the legacy systems is needed to re-modularize the existing assets for supporting service functionality.

The middle-out view consists of goal-service modeling to validate and unearth other services not captured by either top-down or bottom-up service identification approaches. It ties services to goals and sub-goals, key performance indicators, and metrics.

Service classification or categorization. This activity is started when services have been identified. It is important to start service classification into a service hierarchy, reflecting the composite or fractal nature of services: services can and should be composed of finer-grained components and services. Classification helps determine composition and layering, as well as coordinates building of interdependent services based on the hierarchy. Also, it helps alleviate the service proliferation syndrome in which an increasing number of small-grained services get defined, designed, and deployed with very little governance, resulting in major performance, scalability, and management issues. More importantly, service proliferation fails to provide services, which are useful to the business, that allow for the economies of scale to be achieved.

Subsystem analysis. This activity takes the subsystems found above during domain decomposition and specifies the interdependencies and flow between the subsystems. It also puts the use cases identified during domain decomposition as exposed services on the subsystem interface. The analysis of the subsystem consists of creating object models to represent the internal workings and designs of the containing subsystems that will expose the services and realize them. The design construct of the subsystem will then be realized as an implementation construct of a large-grained component realizing the services in the following activity.

Component specification. In the next major activity, the details of the components that implement the services are specified:

- Data
- Rules
- Services
- Configurable profile
- Variations

Messaging and events specifications and management definition occur at this step.

Service allocation. This consists of assigning services to the subsystems that have been identified so far. These subsystems have enterprise components that realize their published functionality. Often you make the simplifying assumption that the subsystem has a one-to-one correspondence with the enterprise components. *Structuring components* occurs when you use patterns to construct enterprise components with a combination of mediators, façade, rule objects, configurable profiles, and factories.

Service allocation also consists of assigning the services and the components that realize them to the layers in your SOA. Allocation of components and services to layers in the SOA is a key task that require the documentation and resolution of key architectural decisions that relate not only to the application architecture but to the technical operational architecture designed and used to support the SOA realization at runtime.

"A huge demand exists for the development and implementation of SOAs"

Service realization. This step recognizes that the software that realizes a given service must be selected or custom-built. Other options that are available include integration, transformation, subscription and outsourcing of parts of the functionality using Web services. In this step you make the decision as to which legacy system module will be used to realize a given service and which services will be built from the "ground-up". Other realization decisions for services other than business functionality include: security, management and monitoring of services.

In reality, projects tend to capitalize on any amount of parallel efforts to meet closing windows of opportunity. Therefore, I recommend conducting three streams in parallel.

Top-down domain decomposition—process modeling and decomposition, variation-oriented analysis, policy and business rules analysis, and domain specific behavior modeling using grammars and diagrams—is conducted in parallel with a bottom-up analysis of existing legacy assets that are candidates for componentization (modularization) and service exposure. To catch the business intent behind the project and to align services with this business intent, goal-service modeling is conducted.

NEW!

CodePro AnalytiX™

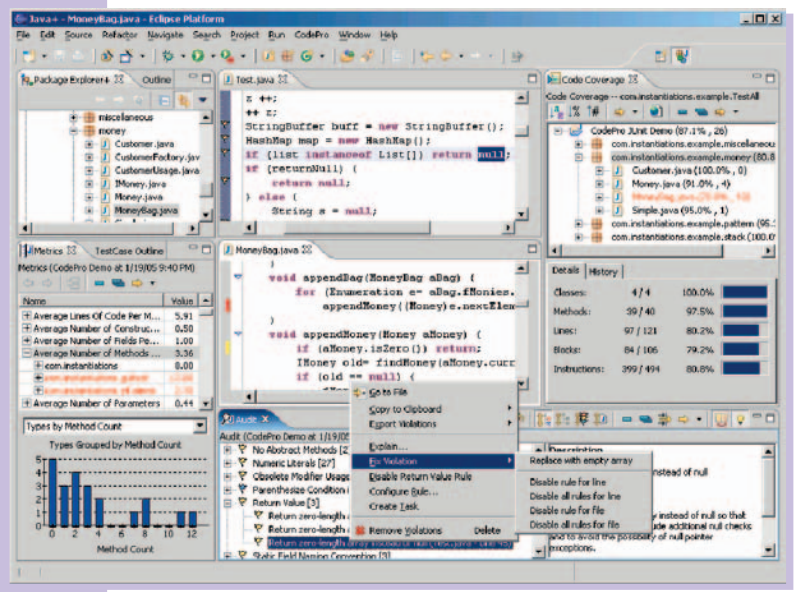
Continuous Collaborative Code Analysis for Rational®, WebSphere® & Eclipse

Imagine your software team creating higher quality, more reliable, Java systems... with less time and money.

Sound like a fantasy?

...*CodePro AnalytiX* lets you do exactly that, or

Your money back.



***AnalytiX* automates the process of building quality into your software as it's developed. Using the latest software development technology *AnalytiX* monitors Java code development in real-time, as its written. It identifies and automatically fixes hundreds of issues that would reduce quality and reliability.**

Powerful wizards generate unit and regression tests and analyze code coverage, helping drive out problems early in the development process.

Less experienced developers code like experts, experts create tight, maintainable systems.

***AnalytiX* also allows you to define, distribute, and enforce corporate coding standards and quality measures across your development teams...no matter where they're located.**

Early identification and correction of code quality issues dramatically improves your application development process and reduces the time and money it takes to develop your high performance Java systems.

From the development tool experts at Instantiations. No one outside of IBM has longer or deeper experience in creating Eclipse-based Java development tools.

Put our experience to work for your team.

www.instantiations.com

1-800-808-3737

Key Product Capabilities

- **Detect and correct code quality issues ...*Automatically***
- **Define, distribute and enforce code quality standards across development teams**
- **700+ rules and metrics/300+ Quick Fixes**
- **Code metrics with drilldown and triggers**
- **Custom rules and rule sets**
- **JUnit test case generation**
- **Code coverage analysis**
- **Dependency analysis and reporting**
- **Integrated team collaboration**
- **Javadoc analysis and repair**
- **Seamless integration with IBM Rational Developer, WebSphere Studio and Eclipse**

Risk Free Trial Copy

www.instantiations.com/analytix

eclipse READY · IBM Business Partner · eclipse FOUNDATION MEMBER

instantiations

Use IBM Rational Application Developer to make life easier

Simplifying Java and J2EE Development

BY STEPHANIE PARKIN & GARY CERNOSEK

Most IT shops recognize the need for applying Java technology. So what keeps Java from becoming even more prevalent than it is today?

Stephanie Parkin currently works as an information architect on the IBM developerWorks WebSphere Web site. She has co-authored two books on visual programming: Building Applications with WebSphere Studio and JavaBeans and VisualAge for Java for Non-Programmers. She is currently writing a book on developing Web services with the Rational Developer products. stephp@us.ibm.com

Most IT shops recognize the need for applying Java technology. So what keeps Java from becoming even more prevalent than it is today?

Some shops are hampered by a lack of in-house Java development skills or the perception that Java is difficult to learn. Others are attracted to the multiplatform benefits of Java but are daunted by the complexities of accessing and deploying to heterogeneous systems. Fortunately, the latest advances in Java technology have significantly reduced the complexity in developing Java applications and making them accessible across the enterprise.

IBM Rational has incorporated many of these advances into its latest product offerings, opening the Java door to many more development professionals. This month we highlight IBM Rational Application Developer for WebSphere Software and how it makes Java and J2EE development easier. (Rational Application Developer is the latest version of WebSphere Studio Application Developer, renamed as part of IBM's ongoing consolidation of development products and resources into the Rational brand.)

Taming Java Application Development Challenges

The majority of developers considering or actively using Java today fall into three basic sets: novice Java developers, legacy procedural developers, and experienced Java and J2EE developers. This section describes these users in detail and the challenges that each user set faces when developing Java applications.

Novice Java developers. Many companies today are experiencing a shortage of skilled Java and J2EE developers. Because these professionals are high-priced, most companies need their existing employees to learn Java skills quickly. Developers trained in client/server programming technologies such as Microsoft Visual Basic are scrambling to get up to speed on the Java language, especially J2EE technology.

Visual Basic was quickly adopted by masses of developers because of the simplicity and ease of use offered by these development tools based on intuitive point-and-click visual composition and object instantiation. Traditionally, Java technology has not offered an equivalent development approach, making the transition to the Java language difficult for Visual Basic developers.

Rational Application Developer provides the ideal environment for novice Java developers, especially those developers familiar with Visual Basic. The new JavaServer Faces (JSF) technology in Rational Application Developer, coupled with its visual and diagram editors, and a data access framework based on Service Data Objects (SDOs), make it easy to develop Web applications without writing code.

Rational Application Developer uses perspectives and progressive disclosure of capabilities to tailor the user interface to the task of the developer. While the product contains a wealth of features for expert developers, it effectively hides the complexity from less experienced users. The wizards, code assist features, integrated best practice guidance, and interactive guides (cheat sheets) all guide new users through the application development process. Rational Application Developer is the ideal tool to get your staff up to speed on Java technology.

Legacy procedural developers. Many companies have procedural programmers who possess a wealth of knowledge about the company's business, legacy systems and databases, but who lack the object-oriented skills required to develop in modern programming languages.

FEATURE	USAGE
Page Designer	Create the visual layout and design of dynamic HTML and JSP pages
Web Site Designer	Manage the structure and navigate entire Web sites
JavaServer Faces support	Quickly develop rich Web user interfaces
Service Data Objects support	Provide a single API for interacting with multiple data sources and visual tools for quickly developing data-driven Web applications
Web Diagram Editor	Map and construct Struts-based Web applications visually
Enterprise Generation Language	Generate Web applications without coding in Java
Portal tools	Visually develop portlets, portals, themes and skins using JSF and Struts
Integrated Web services tools and wizards	Discover and use existing Web services and build, test and publish new Web services
Visual Editor for Java	Extend Eclipse Visual Editor by providing visual tools for binding data sources to controls. Data sources can be EJBs, Web services or JavaBeans
J2EE and EJB wizards and editors	Extend Eclipse by providing wizards, editors and builders to automate creation, testing and deployment of J2EE apps and EJB components. Also supports Xdoclet annotations for rapid deployment
Integrated IBM WebSphere Application Server and IBM WebSphere Portal unit test environments	Provide for unit testing of J2EE and portlet applications
Crystal Reports framework	Design graphical data reports and embed them in Web-based applications
Performance profiling tools	Extend Eclipse by providing thread analysis, additional execution performance views, custom runtime analysis probes and advanced memory leak detection
Component test tools automation	Extend Eclipse by providing structural, complexity and coverage metrics to help decide what to test next, data-pool driven testing, and Web service test generation from Web Services Description Language (WSDL) files
XML tools	Create, edit and transform XML documents
Relational Schema Center	Manage and access databases
Integrated UML Visual Editor for Java, EJB and data tables	Visualize and manage complex code
Code analysis tools	Continuously help ensure code quality and completeness

TABLE 1: IBM RATIONAL APPLICATION DEVELOPER PROVIDES FEATURES BEYOND THOSE FOUND IN ECLIPSE ITSELF

These developers are proficient in COBOL, RPG, C or other procedural languages, and might also be mainframe developers well-versed in subsystems like IBM CICS or IBM IMS.

Fourth-generation (4GL) programming languages bridge the gap between procedural programming and object-oriented programming. They present developers with a familiar programming paradigm and then generate the required deployment code in a different language. IBM provides a 4GL called Enterprise Generation Language (EGL) that generates Java code for execution on IBM WebSphere Application Server. EGL is a simple procedural language—easy to learn for any programmer proficient in business-oriented languages. Developers write their business rules in EGL, and the language automatically integrates that logic with JavaServer Faces components, Struts-based Web applications, the creation and consumption of Web services, and access to message queues, databases, and legacy systems.

The EGL language in Rational Application Developer can help companies modernize their business systems without requiring programming teams trained in procedural languages to master the complexity of the J2EE platform.

Expert Java and J2EE developers. The third set of developers is already proficient in Java, J2EE and Web services technologies. These developers are hindered by the tedious and repetitive coding of low-level programming interfaces that have nothing to do with the application business requirements. These developers need a tool that automates much of the administrative programming so that they can concentrate on implementing business logic that solves unique problems.

Expert J2EE developers often spend the bulk of their time in more critical areas, such as ensuring a sound application design, verifying application performance and throughput requirements or resolving the most elusive and obscure application failures. A tool that automates these tasks can significantly boost the productivity of expert developers.

Rational Application Developer brings together tools for rapidly developing and deploying all the

Gary Cernosek is currently a market manager for the Rational software brand within IBM Software Group. He is responsible for analyzing and responding to software development market trends with a focus on software design and development technology. Gary's experience has been in the visual modeling and model-driven development areas and is now responsible for marketing IBM's newly named Rational Application Developer IDE products. Gary previously held positions in Rational field sales, field technical training, and customer consulting.
gcernosek@us.ibm.com

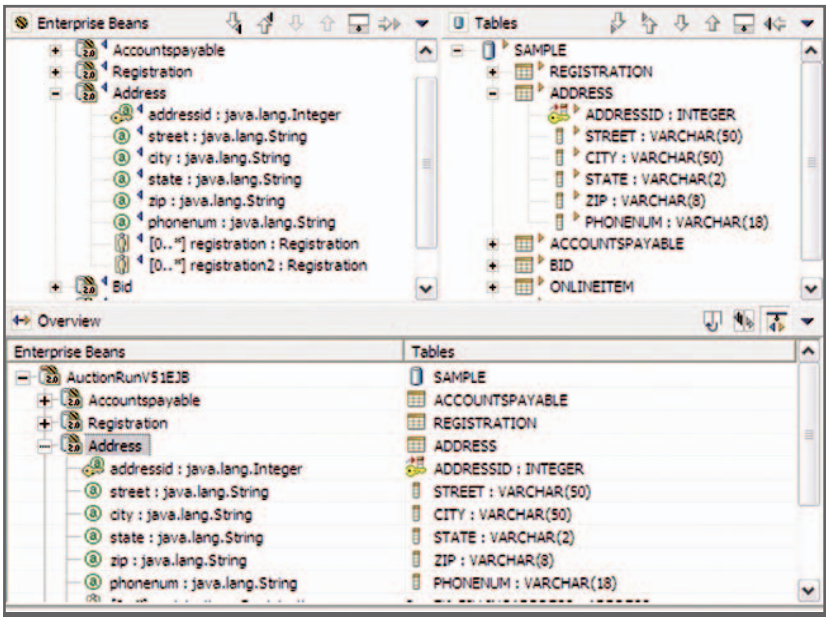


FIGURE 1: THE MAP BROWSER PROVIDES TIME SAVINGS IN CREATING AND MAPPING EJB COMPONENTS

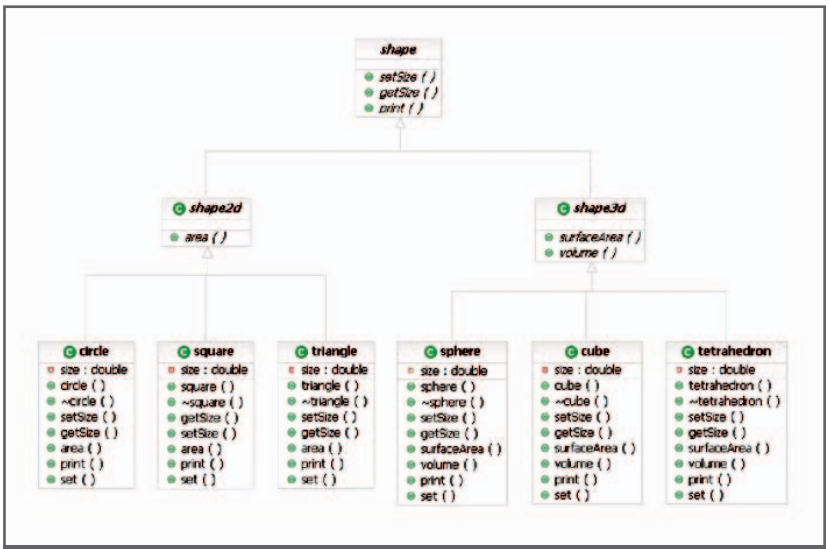


FIGURE 2: THE BROWSE DIAGRAM HELPS DEVELOPERS DOCUMENT THEIR IMPLEMENTATION

various components that comprise J2EE applications: JavaServer Pages (JSPs), servlets, Enterprise JavaBeans (EJB) components, portlets, Web services and SQL queries. It provides Unified Modeling Language (UML) tools to help visualize and understand the structure of complex applications and a comprehensive set of testing tools to help with every step of quality assurance, from unit and remote system testing and automated test case generation to dynamic application performance analysis.

Large organizations with a mix of the three skill sets discussed need a development environment that the entire team can use for collaborating on development projects. They also need a tool that allows for specialization so non-programmers, such as user interface (UI) designers and information developers, can quickly develop their artifacts in the same environment as the rest of the development team. Because it provides tooling for all these different roles, Rational Application Developer speeds up the entire application development cycle.

Extending the Eclipse Platform

Eclipse is an open source, Java-based, extensible development platform for tools integration. Eclipse-based tools give developers the freedom of choice in an environment supporting multiple languages, platforms, and vendors. Eclipse delivers a plug-in–based framework that makes it easy for your team to create, integrate and use software tools together.

Rational Application Developer is IBM's core application development offering built on top of Eclipse 3.0. As a commercially sold and supported product, Rational Application Developer includes many features not available in Eclipse. But because it is built on Eclipse, development teams using Rational Application Developer can adapt and extend the development environment with best-of-breed plug-in tools from IBM, IBM Business Partners, and the Eclipse community to match their needs and maximize developer productivity.

Table 1 gives a quick overview of the features that Rational Application Developer provides beyond those included in Eclipse 3.0.

A Complete Family of Tools

Various configurations of the core technology found in Rational Application Developer help ensure that your IDE scales with your company's needs. As your application requirements grow from simple Web applications to complex, integrated, cross-enterprise business solutions, your developers' skills are preserved and your investment remains protected. Here the family of offerings from IBM that are based on the Rational Application Developer core technology:

- Rational Web Developer for WebSphere Software (formerly WebSphere Studio Site Developer)

is an entry-level IDE for Web and Java developers and primarily used for building JSP and servlet-based Web applications, Java applications and Web services. It supports visual development with JavaServer Faces components and EGL for generating Java code.

- Rational Application Developer for WebSphere Software (formerly WebSphere Studio Application Developer) allows for more advanced J2EE development, including Enterprise JavaBeans (EJB) components. It supports portal and UML-based development, and contains IBM Rational ClearCase LT for version control. Another version of Rational Application Developer is the WebSphere Studio Application Developer Integration Edition, which helps enable accelerated development and integration of composite business applications that deploy to the IBM WebSphere Business Integration Server Foundation. It provides a broad portfolio of rich application and technology adapters and J2EE-based visual workflow tools.
- WebSphere Studio Enterprise Developer adds support for COBOL and PL/1 development and for the development of applications that target legacy back-end systems such as CICS and the IBM zSeries family of servers. It also provides EGL code generation that outputs COBOL source code.
- Rational Software Architect includes all of the capabilities of Rational Application Developer and adds support for UML 2 modeling, patterns, model transforms and code generation, C/C++ development and Java application structural review and control.

While the Eclipse platform does serve as the open source IDE upon which these products are all built, it also serves as a platform for building a wider range of application development tools. As important as design and construction are to the software development process, all significant projects have concerns that span to other facets of the software lifecycle.

That is why the products above are but part of the IBM Rational Software Development Platform, a set of integrated tools, best practices and services that support a proven end-to-end process for the application development life cycle. Rational Application Developer fits into a tools framework that supports team-based application development, including modeling, proven design practices and patterns, and an iterative development process that helps ensure that applications meet user requirements.

Additional Resources

This article is an excerpt from a more complete exploration of Rational Application Developer, a whitepaper titled, "Rapid Java and J2EE Development with IBM WebSphere Studio and IBM Rational Developer." The whitepaper also covers other facets of applying Rational Application Developer, including web, web services, portal, and portlet development. You can download this paper at the link below:
http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/wp-radrwd-medres.pdf

You will find details on how Rational Application Developer can help you accomplish many tasks:

- Visually program Java applications
- Speed up coding of Java and J2EE applications
- Rapidly develop data-access code
- Visualize and edit code with UML
- Automate the build process
- Rapidly deploy J2EE applications

Additionally, we've provided a couple of screenshot here to illustrate some of the ways Rational Application Developer speeds up Java development.

Try It

You can also visit the IBM product trials page and download an evaluation copy of Rational Application Developer to start exploring this rich development environment today:
http://www.ibm.com/developerworks/downloads/index.html

A look at WebSphere Portal Server 5.1

XML Access 101

BY CHRIS LOCKHART

Chris Lockhart has been working with WebSphere and it's various related products since the late 20th Century. For about 6 years he's been working with IBM's WebSphere, Tivoli and Lotus software platforms. Main expertise ranges from performance tuning to AppServer to more obscure things like Edge Server. Just don't call him adeveloper.
chris.lockhart@perficient.com

The world of WebSphere Portal Administration and configuration can be full of tedium and frustration. If you've ever manually constructed trees of portal places and pages, complete with containers and rows and portlets, you may have initially been quite satisfied with yourself. But the satisfaction may have turned rapidly into complete frustration when the portlet you painstakingly set permissions failed to appear when and where you expected it to!

Is it an inheritance problem? Is my portlet activated? Will I meet the deadline for duplicating all of this in both QA and Production? Luckily, WebSphere Portal ships with several useful tools for automating configuration and repetitive administrative tasks. One of the most useful and quickest to understand is XMLAccess.

No-Sweat Solution

XMLAccess is a command-line batch-processing utility that is incredibly useful for exporting and importing various portal configurations. It can be used for backing up the configuration of certain environments, for loading new configurations (for new portlets or pages, for example), or for updating existing portlets when a new WAR file is provided by development. Perhaps most important, this tool greatly eases the duplication of a portal environment from server "A" to server "B." For example, do you need 20 identical Portal Unit Test environments for that new development team? No sweat with XMLAccess.

There are two basic requests that can be made of this interface.

Export requests – tells the utility to export complete or partial portal configurations. This makes no *changes*, it only retrieves the specified configuration and spits it out to an XML file.

Update requests – Yep, you guessed it. This request takes a specified XML file and imports it into the portal configuration database. This process modifies the configuration of the portal based on the values in the imported XML file.

XMLAccess connects to the portal configuration URL over HTTP (usually something like:
http://portal.example.com/wps/config

It therefore can potentially be executed from your workstation or from your home PC, so is very useful for those "working from home" days.

Straightforward Syntax

The XMLAccess command line syntax is very straightforward. For example, if you're on a Windows machine, with a portal was installed at C:\WebSphere\PortalServer, then of course the utility is located on your portal server at C:\WebSphere\PortalServer\bin.

To execute it, you must provide some key bits of info. First specify which XML file you're sending to the utility for instructions on what it should do:

```
xmlaccess.bat –in C:\temp\input-file.xml
```

Next, tell the utility which portal config URL it should connect to in order to execute the instructions contained in that import.xml file. Give it the WPS Admin user ID and password in order to make the changes:

```
xmlaccess.bat –in C:\temp\input-file.xml –user wpsadmin –pwd wps-adminpwd –url http://portal.exam-ple.com/wps/config
```

Finally, tell the utility where to direct the output of your action:

```
xmlaccess.bat –in C:\temp\input-file.xml –user wpsadmin –pwd wps-adminpwd –url http://portal.exam-ple.com/wps/config -out C:\temp\outputfile.xml
```

Voila! When you execute this command, you'll get an output file containing the information you asked for in the inputfile.xml file. Easy, huh?

Export Entire Portal Config

Now that you know the basic syntax, let's look at the very first task in learning to use XMLAccess, which is to produce an export of an existing portal server configuration. This is a very simple XML file construct.

The output of a full export will contain *everything* that is configured in the portal. This will include all portlets, all pages, all permissions, etc. To produce a full export, you first need the XML file that will instruct XMLAccess to generate this export:

```
<?xml version="1.0" encoding=
  "UTF-8"?>
<request
  xmlns:xsi="http://www.w3.org/
    2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "PortalConfig_1.3.xsd"
  type="export"
  export-users="false">
  <portal action="export"/>
</request>
```

This is a very simple XML file with the standard header/schema info. The meat here is the portal element with an action of export. There is a sample of this file on your portal server. It's called C:\WebSphere\PortalServer\doc\xml-samples\export.xml

Create this xml file in a text editor or use the sample. Let's call this file C:\temp\export.xml

Change directory to C:\WebSphere\PortalServer\bin and execute the following:

```
xmlaccess.bat –in C:\temp\export.
xml –user wpsadmin –pwd wpsadmin-
pwd –url http://portal.example.
com/wps/config -out C:\temp\full_
exported_config.xml
```

This will produce a file in C:\temp called full_exported_config.xml containing the entire portal repository configuration. View it in Wordpad or Textpad and you will see all of the components defined to this portal. Once complete, you should see the following message:

```
<status element="all" result="ok">
```

If this export fails for some reason, the failure will be output to the console. Generally speaking, this task will fail if the user/password info is incorrect or if the portal is unavailable—or if you're using the wrong port in the –url section).

Export Partial Portal Config

But what if you don't want to export the entire configuration of the portal? Let's say you only want to export the configuration of your new page with only the components you placed on that page.

Your page is a "content-node" in XMLAccess parlance. If you look at the full export you did above, toward the bottom you'll see all of the default out-of-the-box content-nodes defined in that file (Welcome, My Work, My Finances, etc).

Each default content-node has a unique name defined for it in the portal config. Any new content-nodes that you may add (such as newpage1) will not, by default, have a consistent uniquename. It is a good idea to define one for your new page before we continue. This will make it much easier to locate your page in the config file.

Let's say your page is called newpage1. You created it in the Manage Pages section of the portal admin console and you placed it under Content Root->My Portal. That is, it is at the same level as the default 'Welcome' page.

"It's a good idea to define custom uniquenames for your new pages"

Give this new page a unique name by going to Portal Settings->Custom Unique Names in the portal admin console. Select Pages as the Resource Type, find your newpage1 (it's probably on the very last page of the list) and click the pencil icon to edit the unique name.

Let's give newpage1 the unique name of wps.My Portal.newpage1. From now on, instead of referring to your page as some enormous (and changing) value in the XML, we can locate it by this name.

To export just your newpage1 configuration, you would create a new XML file. Let's call this one newpage1_export.xml. This new file would contain the following:

```
<?xml version="1.0" encoding=
  "UTF-8"?>
<request
  xmlns:xsi="http://www.w3.org/
   2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "PortalConfig_1.3.xsd"
  type="export">
  <portal action="locate">
    <content-node action="export"
      uniquename="wps.My Portal.
      newpage1" export-
descendants="true"/>
  </portal>
</request>
```

Notice we're only "locating" the portal configuration initially, whereas before we exported it. Once we locate the portal config, we drill in further and do an export on the content-node named wps.My Portal.newpage1.

Execute this as follows:

```
xmlaccess.bat –in C:\temp\new-
page1_export.xml –user wpsadmin
–pwd wpsadminpwd –url http://por-
tal.example.com/wps/config -out
C:\temp\partial_exported_config.
xml
```

We now have just your newpage1 configuration exported to C:\temp\partial_exported_config.xml.

So simple. Following this methodology (and using the helpful sample files in C:\WebSphere\PortalServer\doc\xml-samples), we can export a single portlet, a theme configuration, permissions, etc.

Import Full Portal Config

Now that you have your exported config, perhaps you'll want to import it into a fresh new portal. Let's pretend that we have a portal on a new Windows machine and that it was created with

"The actual WAR files and themes and skins files are not part of these exports/imports. You must move these pieces manually"

an empty configuration, using the -W installPortletsSequence.active="false" option. (For more info on how to do this, consult the Portal InfoCenter).

There are no pages or labels or portlets defined with this blank portal. You first take your full export (generated earlier), and transfer the file over to the new portal. Stick this file in C:\temp.

Remember, you only have the desired *configuration definition* in your full_exported_config.xml file. The actual WAR files and themes and skins files are not in the XML file. You have to make sure that any portlets or themes or skins that are defined in this config file have corresponding physical files on the server. For example, if you have a definition for portletabc.war in our config file, then you have to ensure that portletabc.war is actually located on the server we plan to import the config on.

By default, the portal will assume that portlet WARs will be located in C:\WebSphere\PortalServer\installableApps. The portal will likewise assume that files for the themes and skins are located in the default locations.

Once you verify that all the necessary portlet WARs and theme files are in place, you can execute the import:

```
xmlaccess.bat –in C:\temp\full_
exported_config.xml –user wpsadmin
–pwd wpsadminpwd –url http://por-
tal.example.com/wps/config
```

Note that you don't *necessarily* have to specify an output file. After all, you're importing a config in this step. If you specified an output file using the –out flag, the console output would be written to this file. Once you execute this and restart your portal, you should see the entire config from portal A now transferred to your new portal B.

Import Partial Portal Config

If you are importing the partial config generated above, you need to be aware that you're only importing that single newpage1 and the pieces contained under it. You're not importing the parent content-node for our newpage1, but instead assuming it is already there.

When you exported the partial config, you produced: C:\temp\partial_exported_config.xml that should look something like this:

```
<?xml version="1.0" encoding=
  "UTF-8"?>
<request
type="update" xmlns:xsi=http://
  www.w3.org/2001/XMLSchema-
  instance xsi:noNamespaceSchema
  Location="PortalConfig_1.3.xsd">

  <portal action="locate">
<skin action="locate" objectid="_
K_0830MR0T4A0I1P9R_37 Outline"
uniquename="wps.skin.outline"/>
<theme action="locate" objectid="_
J_0830MR0T4A0I1P9R_3B WebSphere"
uniquename="wps.theme.webSphere"/>
<content-node action="locate"
objectid="6_0_A Content Root"
 uniquename="wps.content.root"/>
<content-node action="locate"
objectid="_6_0830MR0T4A0I1P9R_35
My Portal"
 uniquename="wps.My Portal"/>

<content-node action="update"
active="true" allportletsallowe
d="true" content-parentref="_6_
0830MR0T4A0I1P9R_35 My Portal"
create-type="explicit" objectid="_
6_0830MR0T4A0I1P9R_CI newpage1"
ordinal="900" skinref="undefined"
themeref="_J_0830MR0T4A0I1P9R_
3B WebSphere" type="page"
uniquename="wps.My Portal.new-
page1">
    </content-node>
  </portal>
  <status element="all"
    result="ok"/>
</request>
```

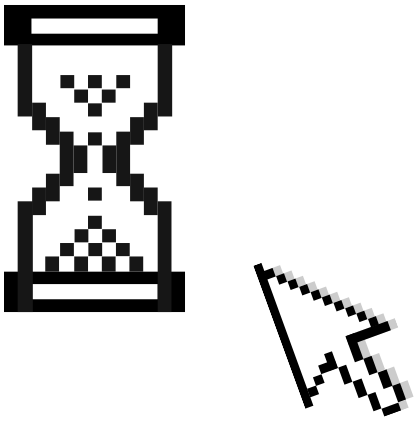
You can see how you're running a "locate" on the Content Root content-node and the My Portal content-node. If those nodes don't already exist in the new portal, then the partial import of newpage1 will fail. The same is true for the skins and themes and any portlets that may be defined on our newpage1.

If all the objects that you're "locating" are found, then your newpage1 will be created in the appropriate spot when you run the import on portal server 'B':

```
xmlaccess.bat –in C:\temp\partial_
exported_config.xml –user wpsadmin
–pwd wpsadminpwd –url http://por-
tal.example.com/wps/config
```

Log into and you should see newpage1. Brilliant!

As you tinker with importing and exporting, you'll discover how easy it is to update access permissions, add or update portlets, add or update themes, skins and the like. The example files in C:\WebSphere\PortalServer\doc\xmlsamples are your starting point for these other activities. Also of great use are the InfoCenter documents regarding XML Access. These pages are currently located at http://publib.boulder.ibm.com/infocenter/wp51help/topic/com.ibm.wp.ent.doc/wps/admxlai.html.



World Wide Wait.

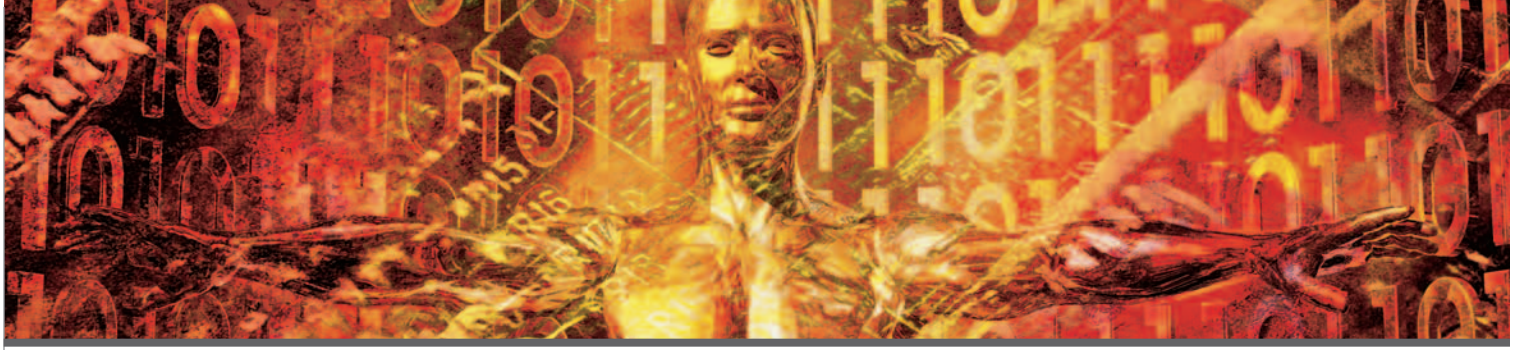
Don't Leave Your Customers Hanging — Ensure a Positive End User Experience with Quest Software.

They don't care where the problem is. All that matters is their experience using your site. But can you quickly detect and diagnose a problem and deploy the right expert to fix it?

Quest Software's solutions for J2EE application performance management shed light on the true end user experience and help you isolate and fix problems across all tiers. Whether a problem lies in your server cluster, database, framework or Java code, nothing gets you to resolution and optimal performance faster.

Don't just cross your fingers and hope for the best. Get more with Quest.

Download the free white paper,
"Measuring J2EE Application Performance in Production"
at www.quest.com/wsdj



"Server, Heal Thyself"

Autonomic features in WebSphere v6: Peer Recovery of Transactions

BY DENNIS W. ASHBY

Dennis W. Ashby is a senior J2EE architect and WebSphere developer who specializes in server-side and back-end systems. Having joined Prolifics in 1996, Dennis is a key player in Prolifics' highly specialized group of WebSphere consultants. Headquartered in New York City, Prolifics is retained by IBM to apply their specialized WebSphere skills at any phase of the customer's project life cycle. Dennis has applied Prolifics' proven methodology and expertise in the design of J2EE applications and service-orientated architectures to clients all over the world including Pep Boys, Citigroup, KeySpan Energy, Allmerica, American Express, FEMA, Bank Zachodni, and Gelco.
dashby@prolifics.com

At the forefront of IBM's autonomic computing strategy is WebSphere Application Server v6. This version of the application server is designed to automatically detect problems ranging from small network glitches to large-scale power failures and, in a matter of seconds, save and process Web-based business transactions that could take hours or days to recover under older systems. This is referred to as "self-healing" software.

As soon as the application server detects a problem, the transaction and related data is automatically moved to another server either within the same data center or, in the case of a power outage or disaster, to another geographic location. This new functionality is known as "peer recovery of transactions." This article will discuss how this works and how to configure it in WebSphere Application Server v6.

High Availability Manager

Peer recovery of transactions is tightly coupled with the configuration of WebSphere's high availability environment. So let's start by taking a brief look at the WebSphere high-availability architecture.

WebSphere's architecture for high-availability employs the use of a high-availability manager (HAM) to monitor key services provided by the application server. These services encompass messaging, transaction managers, workload management controllers and other application servers in the cluster. Today, many enterprise application architectures include the use of network-attached storage (NAS) devices. A requirement of peer recovery is that the HAM makes use of such a device for storage of transaction logs from each application server in the cluster.

The HAM is responsible for automatic recovery of all in-flight as well as in-doubt work for any application server that fails in the high-availability cluster. This automatic functionality allows the WebSphere cluster to re-stabilize itself if one or more of the cluster members should fail.

In WebSphere v6, all of the application servers in a cell are defined as members of a core group. A core group is a statically-defined set of application servers that can be divided up into various high-availability groups. Each core group has only one logical high-availability manager that continually polls all of the group members to verify they are active. It is also responsible for making the services within the core group available and scalable.

WebSphere uses policy matching to localize and partition policy-driven work into high-availability groups. In other words, when a core group member (in this case, an application server) fails, the high-availability manager can dynamically reassign the failing member's work to another component from the same high-availability group (in this case, a healthy application server). Using NAS devices in the position of common logging facilities, the component which has been assigned the work can recover and process the in-doubt and in-flight work of the failed component. This is essentially the heart of peer recovery of transactions.

Transaction Service Logs

Knowing this, we can examine results. The WebSphere transaction service logs are essential for providing recovery of transactions, enabling transactional work in progress at the time of a server failure to be resolved when the server is restarted. These logs contain information about active transactional work that is preserved across a server crash.

WebSphere v6 allows you to configure the location of the transaction log directory, either using the WebSphere administrative console or commands. Special logic has been added to the administrative console to facilitate the migration of the transaction log configuration from earlier versions of WebSphere. In older versions the transaction log directory configuration was stored in the *server.xml* server-level configuration file. In the current version it is stored in the *serverindex.xml* node-level configuration file.

Prior to WebSphere v6, when an application server

restarted from a crash, the server's recovery processes included retrieving the transaction service logs, processing the recorded information, recovering the transactional work and completing the in-doubt transactions (including the release of database locks). The server had to first completely restart and process the logs before the completion of the transactional work would take place. Thus, if the server was slow to recover or if it required manual intervention, the transactional work could not complete, and access to the database was disrupted.

Depending on the industry, IBM points to estimates that the failure of an e-business application can cost a company as much as $110,000 per minute in lost revenue and productivity. To minimize the cost and disruption of a failure, WebSphere v6 employs the strategy known as peer recovery of transactions.

Peer Recovery of Transactions

The peer recovery strategy is provided only within a server cluster, moving the high-availability requirements away from individual servers and onto the server cluster. As shown in Figure 1, a recovery process runs alongside the standard application server processes, enabling one server in the cluster to recover the work for another failed server in the same cluster. Thus, to recover the failed server, you do not need to start up a new application server.

The peer-recovery process is the logical equal to restarting the failed server, without actually doing so. It is important to note that the recovery process completes outstanding work, but does not start new work or provide "forward processing" functionality. The Workload Manager (WLM) of the cluster can then dispatch new work onto the remaining servers. Both transactions and the compensation service failover together to the same peer server. The only difference from the user's perspective is the potential drop in overall system throughput.

Self-Healing Example

Let's consider an example of the peer recovery process that takes place if a single server fails. Three servers (S1, S2 & S3), as detailed in Figure 2, are running in a clustered WebSphere Application Server environment with the Workload Manager load balancing the work between the servers. Note, database locks are being held for each of application servers.

During normal processing, let's suppose the S1 application server unexpectedly crashes. This will result in locks being held in the database. Subsequently, the application servers S2 and S3 are able to continue processing their existing transactions, but future transactional processing may be impeded due to the locks still held on behalf of crashed S1 application server.

In WebSphere v6, once S1 had failed, a peer recovery process for S1 would begin running inside S3 (or S2 depending on the configuration). The transaction service portion of the recovery process would retrieve the information persisted by S1, and use that information to complete any in-doubt transactions.

In addition, to recover any in-flight transactions, the endpoint references for S1 would be redirected to S3. Thus, the system would remain in a stable state with just two servers, between which the WLM engine can balance workload. At some future point when S1 is brought back online, it will have no recovery processing of its own to perform (see Figure 3).

Configuring the transaction properties required for peer recovery is part of the overall task for configuring a cluster to use high availability support. Note that a cluster can house both v5 and v6 application servers; however, the peer recovery support is only available with clusters where all the members are v6 servers.

To configure the transaction properties required for peer recovery, complete the following steps:

1. Configure the transaction log directory setting for each server in the cluster. You can configure the location of the transaction log directory using either the WebSphere administrative console or commands.
2. For peer recovery, each server in the cluster must be able to access the log directories of other servers in the same cluster. Interactions between the High Availability framework and the recovery log service prevent concurrent access to a single physical recovery log.
3. When using WebSphere Application Server without High Availability support, you can leave the recovery log configuration for persistent services (such as the transactions service) unset. The application server assumes a default location within the appropriate profile directory. When High Availability support is enabled, this default

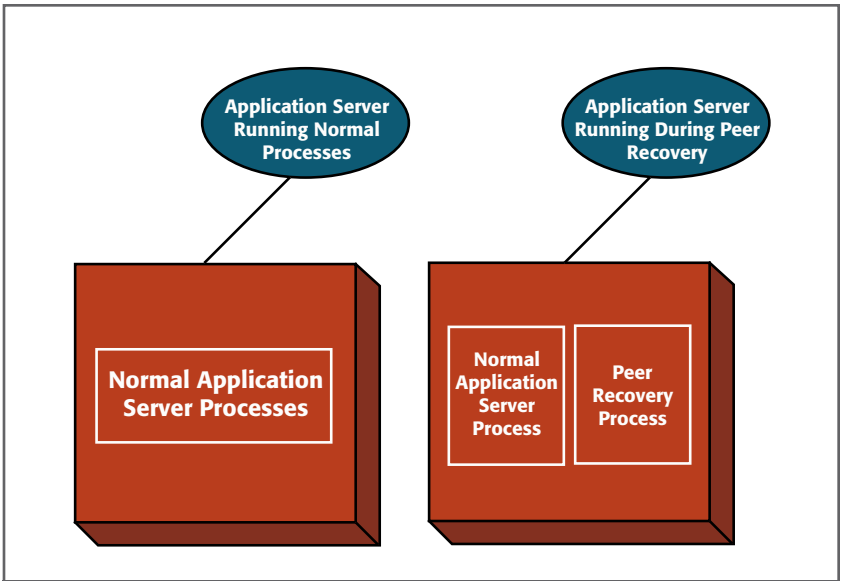


FIG 1: RUNNING THE PEER RECOVERY PROCESS

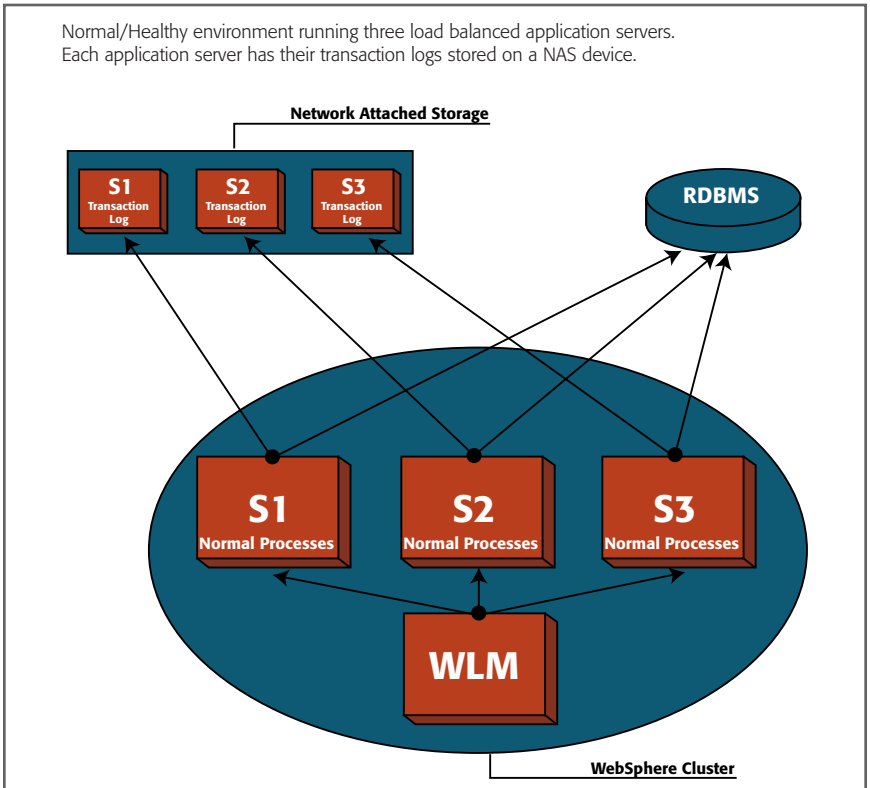


FIG 2: HEALTHY THREE-SERVER ENVIRONMENT

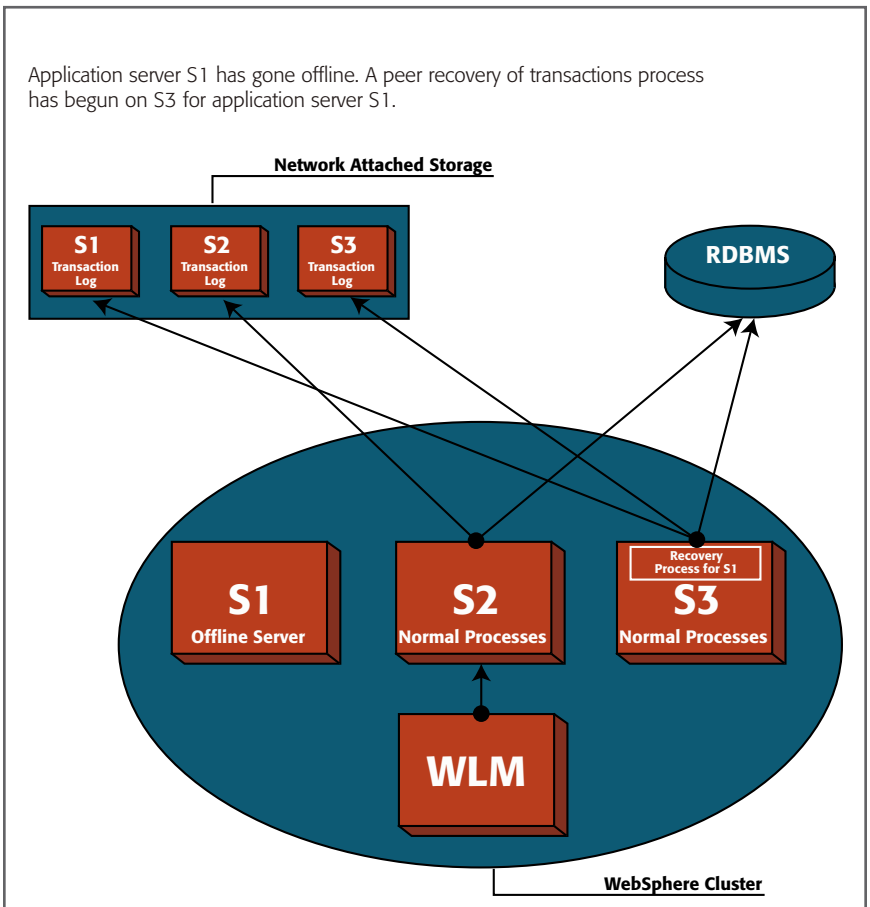
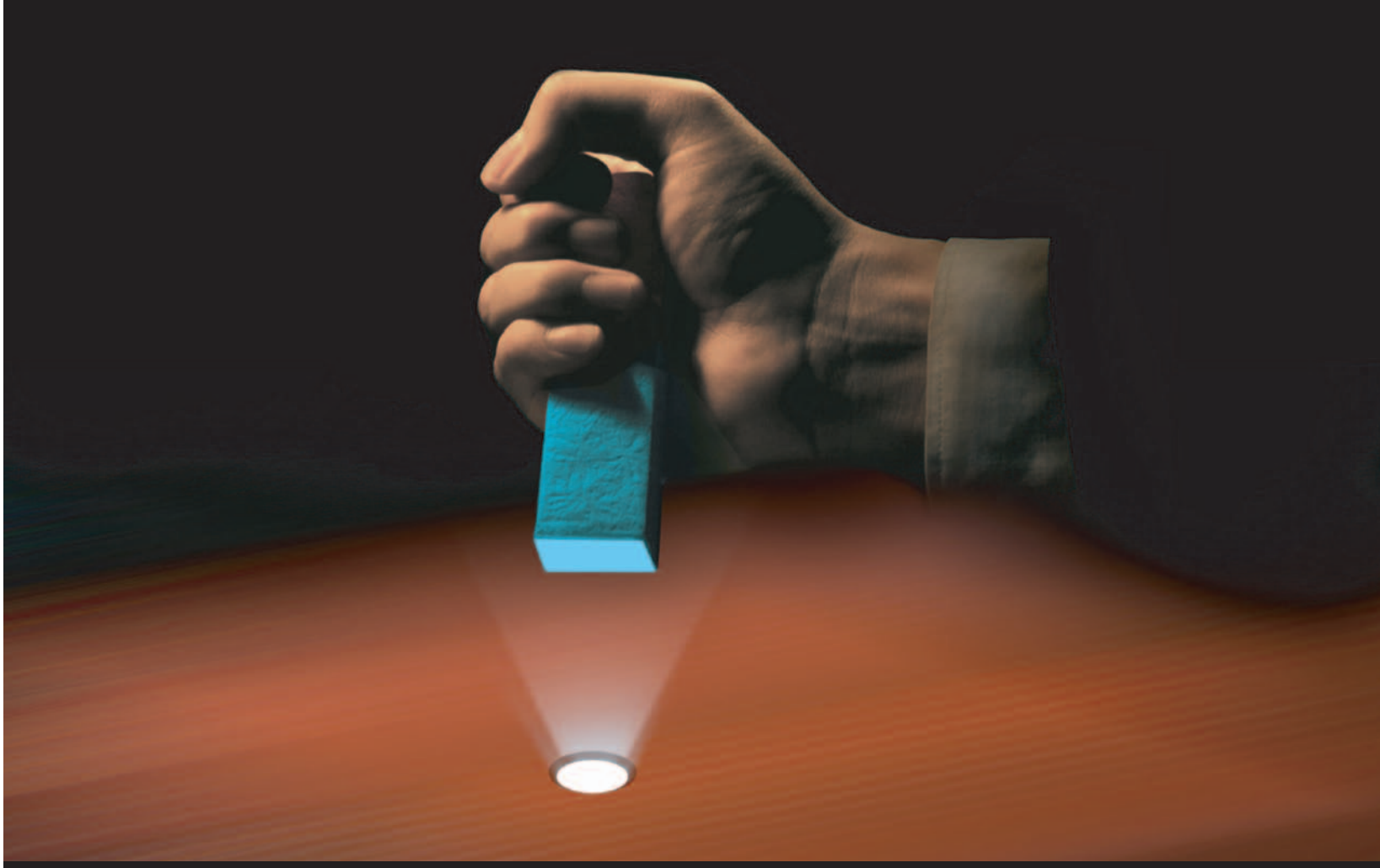


FIG 3: A PEER RECOVERY OF TRANSACTIONS PROCESS BEGINS ON S3

may not be visible from all servers in the cluster (for example, if they are in different profiles or physical nodes.) As a result, it is recommended that the recovery log location be configured for each server in the cluster before enabling High Availability.

4. The storage mechanism used to host recovery log files and access to that mechanism must support the file-based force operation that is used by the recovery log service to force data to disk.
5. Enable the High Availability function for the cluster, by completing the following steps on the cluster configuration panel of the WebSphere administrative console:
6. In the administrative console, click **Servers** > **Clusters** > ***your_cluster_name***.
   a. To enable high availability for a cluster, select the **Enable high availability for persistent services** option.
7. To change the peer recovery characteristics for a specific server in the cluster, you need to create a new policy that defines the criteria for the specific server.
   a. Change the **WAS_TRANSACTION** policy in the core group configuration
8. In the administrative console, click **Servers** > **Core Groups** > **DefaultCoreGroup** > **Policies** > **Cluster TM Policy**.
   a. Change the policy properties to suit your recovery requirements.

Once you have completed these steps you are now ready to take advantage of peer recovery of transactions which is at the center of WebSphere's "self-healing" functionality. By incorporating autonomic computing strategies into WebSphere v6, IBM has taken a giant leap forward toward self-healing, self-configuring, and self-protecting systems. With systems that can immediately begin to process in-doubt and in-flight work when they detect a server problem, your applications can achieve a greater level of reliability and success.



Think Crystal is a good fit for your Java application? Think again.

Think JReport.

Forcing a Windows reporting solution into a J2EE environment is never going to result in a perfect fit. Only JReport is built from the ground up to leverage J2EE standards and modular components for seamless embedding in any Web application.

JReport is ready out-of-the-box, with all the tools necessary to empower your application with any reporting requirement. From reusable, shared report components to flexible APIs, JReport is the most complete embedded reporting solution available.

With the ability to scale to multi-CPU and server clusters, JReport is a perfect fit for any reporting workload. Load balancing and failover protection provide peak performance and uninterrupted access to critical business data. In addition, JReport integrates with any external security scheme for single sign-on.

JReport's ad hoc reporting lets users access and analyze data on demand, from any browser. And, with cascading parameters, embedded web controls for dynamic sorting and filtering, drill-down and pivot, JReport ensures users get the information they want, when they want it, and how they want it.

See for yourself why over half of the world's largest organizations have turned to JReport to enable their J2EE applications with actionable reporting.

When it comes to embedded Java reporting, JReport is the perfect fit. Download a FREE copy of JReport today at www.jinfonet.com/wj4.

JReport
JINFONET SOFTWARE

The Big Picture

How Portal Server and Porlet API fit into WAS

BY POOJA GUPTA & LAURA MENKE

Pooja Gupta is staff software engineer with IBM Software Group. She has seven years of experience in the IT field. She holds a Bachelor degree in Computer Science and is an IBM Certified WebSphere Portal 5.0 Solution Developer. Her areas of expertise include working on WebSphere products and end-to-end development of J2EE architecture–based Web applications. She is a technical writer in internal IBM Redbook. pooja_gupta@us.ibm.com

IBM WebSphere Portal Server is based on J2EE architecture and requires IBM WebSphere Application Server to run. IBM WebSphere Portal Server helps to personalize and aggregate content, and eliminat es information overload for the users. Web site developers want to keep a consistent look and feel to the Web site all across, and to ensure that the most common method is usage of .inc files, aka include files.

The portal could provide a user with a consistent view on portal applications and in turn remove/minimize the cumbersome static management of include files and save time for developers. Portal offers dynamic theme and skin management. Developers can develop a theme, install in Portal Server and can inherit the theme in the required page. Skins are associated with a Theme, and a Theme must have a skin associated with it. If none is desired then the "No Skin" option is selected. Skin can somewhat be related as a background color on a canvas and a theme can be a logo of a company on that canvas. Changes in the theme and skins can be viewed in JSPs without restarting the portal server, since the application server can be forced to automatically check for new versions of JSPs. This is ideal for development and testing purposes; automatic JSP reloading should be disabled in a production environment in order to avoid performance issues.

IBM WebSphere Portal supports HTML, WML, and cHTML markup languages. To support another markup language, you must ensure that you have portlets that provide markup for the language. In addition, it would need to create a subdirectory for the markup in each of the following locations:

was_root/installedApps/*hostname*/wps.ear/wps.war/screens

Screens are JSPs invoked by the portal server for displaying various content. The Home screen is most often displayed because it is used for the portlet content area. Other screens include the Login or Error screen.

was_root/installedApps/*hostname*/wps.ear/wps.war/themes

In addition to JSPs, the themes directory might need images, style sheets, or any other file types required of the new markup.

was_root/installedApps/*hostname*/wps.ear/wps.war/skins

Portal container relies on the J2EE architecture implemented by IBM WebSphere Application Server. As a result, portlets are packaged in WAR files similar to J2EE Web applications and are deployed like servlets. Portlets are reusable components that provide access to Web-based content, applications, and other resources. Web pages, applications, and syndicated content feeds can be accessed through portlets. Portlets are intended to be assembled into a larger portal page, with multiple instances of the same portlet displaying different data for each user.

Portlets offer a unique concept—cloning of the base portlets—where cloned portlets are called "concrete portlets." These concrete portlets share the same base code as the parent or the base portlet though they have different portlet settings and configuration values. This feature not only saves time but also gives developer flexibility of reusing the base portlet code.

Applications in general should follow the MVC design pattern, where M-Model, V- View, and C-Controller.

Model

The model is the core of application's functionality and has no knowledge of the view or controller.

VIEW

The view is the presentation layer of the application.

CONTROLLER

The controller decides where to switch the control of the application when the user initiates.

Figure 1 diagram illustrates portlets being used in an MVC paradigm where portlets act as controller, JSPs serve as view and the back end acts as a model. Numerals in braces are the steps in which the application flows.

Portlets get the request from the browser and its doview method is activated; this method passes the control to JSP. JSP, with the help of JSP tag lib, connects to back end to get the desired XML. XSLT is attached to the resulting XML and transformed to HTML using Xerces parser and displayed to browser. JSP files are a part of WAR that is deployed in the portal server while XSLT either can be served from the IHS or the edge server of the same Web server or different Web server, depending on the architecture of the application.

IBM WebSphere Studio Application Server, very popularly known as WSAD, is the recommended portal development platform. WSAD can be linked to Clearcase/CVS respository for source code management. Xerces or Xalan parsers (open source API) can be used to transform XSL+XML =HTML, since the final result displayed on the browser should be HTML, making it browser-independent.

In case the resulting output is not HTML, then it would only work fine on the Internet Explorer since it has a built-in XSLT, and not on Netscape 4.x. Usage of Portal API defined tag libraries and user defined tag libraries is encouraged in order to encapsulate the core api from front end developers and to reuse the code.

IBM WebSphere Portal Server is an excellent option for Web site developers looking to save time, minimize the management of number of pages, personalize the content, centrally manage site look and feel and better performance of Web sites, though it has its own challanges of skill set and infrastructure cost and maintenance.

"IBM WebSphere Portal Server is an excellent option for Web site developers looking to save time"

Acknowledgments

I dedicate this article to Laura Menke, my technical mentor.

To learn about IBM WebSphere Portal Server Offerings and products in detail, please visit: http://www-306.ibm.com/software/genservers/portal/.

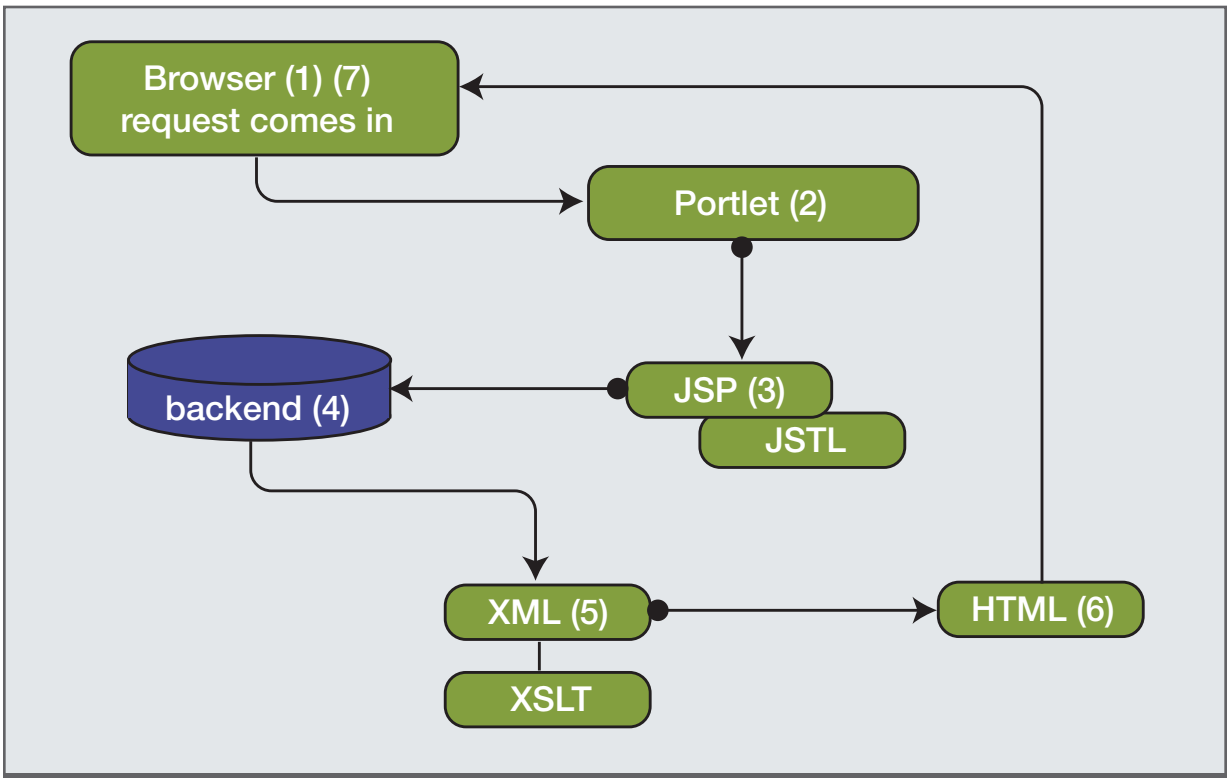
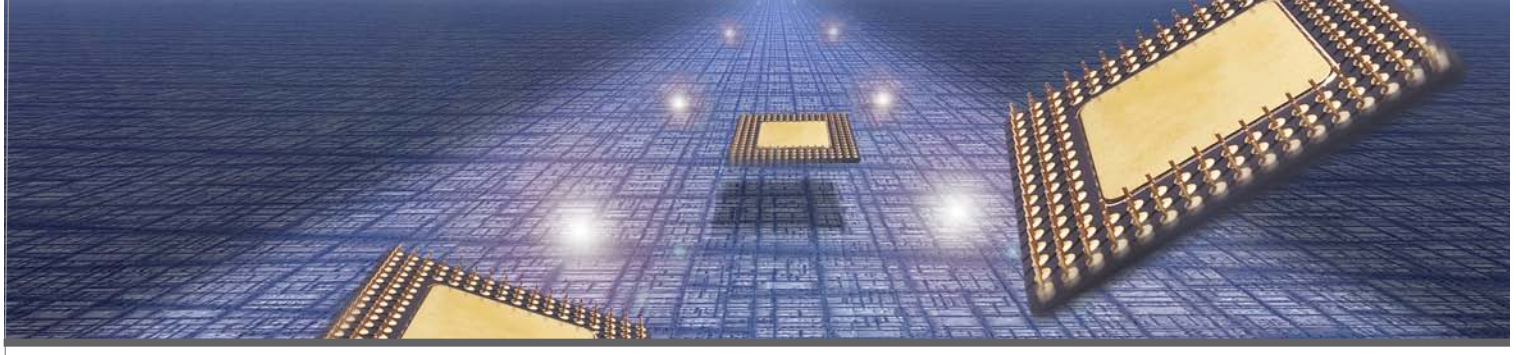


FIG 1: PORTLETS IN AN MVC PARADIGM

Laura Menke is an advisory IT architect at IBM. She has seven years of experience working at IBM on various Java Web application projects. Her expertise is with the WebSphere family of products and is certified on each release of WebSphere Portal. She is serving as an advisory on an internal IBM Redbook . menkell@us.ibm.com



Support of Composite Applications in WebSphere Portal

Part 2 – The meat and potatoes

BY DOUGLASS WILSON & RICHARD GORNITSKY

Richard Gornitsky is a WebSphere Portal architect with IBM's Software Services for Lotus. His expertise is in integrating WebSphere Portal into Fortune 500 firms from concept to production. Richard?s 23 years of industry experience includes finance, insurance, telecommunications, pharmaceutical, software manufacturing, and retail/distribution. He has experience in the full life cycle development of high transaction solutions, which includes simultaneously managing multiple large complex application development projects. Richard is a requested technical speaker and is a coauthor of Wiley Technology Publishing?s Mastering WebSphere Portal.
rsgornit@us.ibm.com

In the first of our three-part series, published in November 2004, we introduced the composite application model and how WebSphere Portal supports it. Composite applications are a class of application whose fundamental construction model is the composition of parts or components. They are formed by choosing components or services from a catalog or palette and interconnecting the components to create the behavior desired.

The foundation of the composition model is the portal infrastructure, which provides access to user profile information and remote content, the ability to communicate and trigger other portlets and to participate in windows and action events, store persistent data, and look up credentials.

Composite applications within WebSphere Portal are built from portlets. WebSphere supports this model by allowing portlets to be composed into composite applications through the same tooling used to design portal pages. Portlets can be considered to be the components in the composition model since they are each self-contained applications that support multiple states and view modes.

While simply exposing business data and operations through a portlet offers a lot of value, even greater value can be obtained when portlets work together. To create a composite application the portlets must be able to pass data to each other, react to changes in their data, trigger events, and respond to events from other components.

Communication and interaction with other composite application components take place through defined interfaces such as messaging, event handling, or services. Each portlet can be removed or added from the model or page without impacting other portlets.

What are Co-operative Portlets?

Composite applications in WebSphere Portals are built on co-operative portlets, which support the trigger/workflow model. Co-operative portlets pass information to each other using the WebSphere Portal property broker, which enables portlets to publish their typed data , or properties, and to declare actions (services) that they are willing to execute based on the data values published by other portlets.

The portlet that provides a property is called the "source portlet," while the properties that the source portlet publishes are called "output properties." The portlet that receives a property is called the "target portlet" and the properties that are received by the target are called "input properties." With cooperative portlets, the target portlet can provide an action based on the property or data it receives.

Each value passed has meta-information associated with it—specifically the type, namespace, and direction. The type specifies the semantics of the property, the namespace provides a unique name to prevent collision, and direction specifies whether it is an input property or an output property.

A co-operative portlet can register its properties programmatically or through the use of WSDL file. If a WSDL file is used, the property broker will parse it and register the actions and properties.

There are two modes of interaction between co-operative portlets: *click-to-action (C2A)* and *wired.* C2A allows end-users to trigger data transfers from the source portlet to the target portlets using pop-up menus that appear next to the data being transferred. The menus are created dynamically by the portlet property broker. Wired co-operative portlets entail the implementation of persistent connections between portlets. A data change in the source

portlet automatically triggers a change on the target portlets without requiring the intervention of the user.

If the target action was specified in the wire, the Property Broker delivers the property values through the *actionPerformed* method. However, if the target property is specified in the wire, the property value is delivered through the *setProperties* method.

Creating Composite Applications with Co-operative Portlets

Creating and enhancing composite applications with co-operative portlets is not difficult! There are two methods—either using the Portlet Wiring Tool, or doing it through programming. We will first show you how to do it the hard way, since that is the way most geeks (like us) like doing it.

Our examples will be based first showing how IBM Workplace for Business Control Reporting (IWBCR) exploits the composite application model through the use of co-operative portlets. IBM Workplace for Business Control Reporting is a WebSphere Portal Application written and marketed by IBM. (IWBCR was created in response to the Sarbanes-Oxley Act of 2002. The act is intended to help protect stockholders of companies with shares trading on U.S. exchanges and to place more accountability on the decision makers within those companies. IBM Workplace for Business Controls and Reporting is a solution designed to help companies manage and assess their business controls. The offering provides corporations with the ability to document, monitor, and test internal controls that can help manage financial reporting processes in a cost effective and sustainable manner. It allows organizations to help drive down the cost of control with features to help manage organizational change, simplify testing and auditing and help improve performance through an update option that ensures different departments have the most current data at their fingertips.)

WBCR portlets take the advantage of co-operative portlets to synchronize the data displayed in multiple portlets when re-acting to user actions on one portlet. IWBCR uses programmable co-operative portlets extensively to enable an action in one portlet to trigger an event in another portlet.

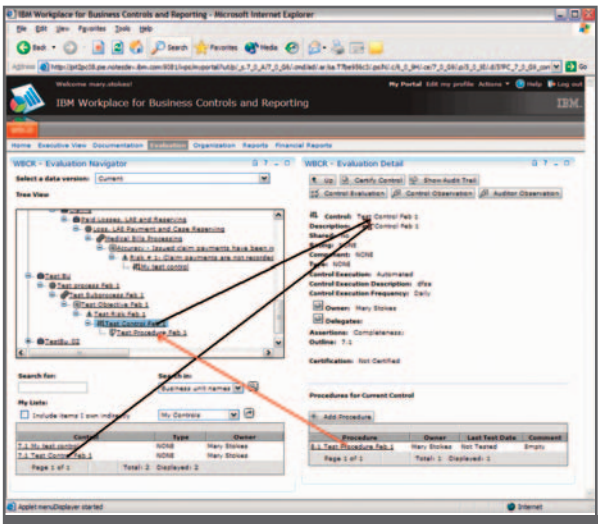


FIG 1: IWBCR EVALUATION PAGE

Figure 1 shows two portlets on the same page and how they interact. When the user clicks on a tree node (Test Control Feb 1) in the Evaluation Navigator portlet, it triggers a co-operative portlet event. The Evaluation Detail portlet receives this event, and displays the details of the item clicked on the tree. If a user clicks on an item on the table in the Evaluation Navigator (Test Procedure Feb 1), it triggers a cooperative portlet event, the Evaluation Detail portlet on the left hand side receives the event, and sets the tree selection on the tree node (Test Procedure Feb 1).

To make this work, you must make sure that the pbportlet.jar (which contains the PortletWrapper class) is in the WEB-INF/lib directory of the war file. Then the co-operative portlets must be wrapped by a special portlet, PortletWrapper. In the *web.xml* file within the WAR file, you need to define the main servlet class, *c2a-application-portlet-class,* specify it again through a special <*init-param*> as follows:

```
<servlet-class>com.ibm.wps.pb.wrapper.
  PortletWrapper</servlet-class>
    <init-param>
  <param-name>c2a-application-portlet-class
    </param-name>
  <param-value>com.ibm.lwp.bcr.portlet.
    BCRPortlet</param-value>
```

In the *portlet.xml file*, we specified the location of the WSDL file;

```
<config-param>
      <param-name>c2a-action-descriptor
        </param-name>
      <param-value>/wsdl/CatTree.wsdl
        </param-value>
</config-param>
```

The portlets register their intent to publish property values and/or receive notification of property changes by implementing methods of the interfaces EventPhaseListener.

```
public class BCRPortlet extends AbstractPortlet
implements ActionListener, PortletTitleListener,
EventPhaseListener
```

We won't describe the methods for registering properties. They are described in the WebSphere Portal Infocenter. http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html.

However, we will illustrate the methods used to actually publish the values and value changes for registered properties.

Doug Wilson is an IBM distinguished engineer and chief technology officer for the Lotus Product Division of IBM's Software Group. Prior to this he was responsible for architectural strategy for Domino and WebSphere integration, WebSphere Portal Server and WebSphere content manager, and WebSphere Personalization. He invented and managed the design and development of Lotus InfoBus technology, a data exchange technology for Java components that has been accepted for inclusion in JavaSoft's Java Developer's Kit as a Java standard. Doug holds a degree from M.I.T.'s Department of Mechanical Engineering.
douglas_wilson@us.ibm.com

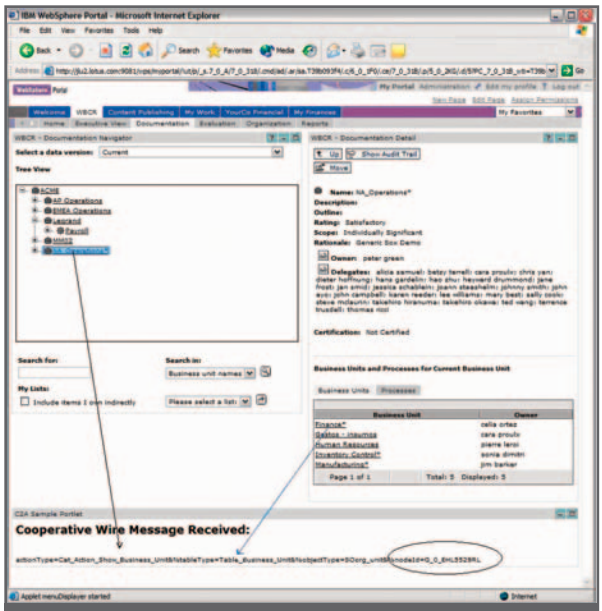


FIG 2: DOCUMENTATION PAGE

Portlets wishing to publish a property value do so by calling methods of the property broker service built into the portal. (In addition to directly calling methods from the property broker, portlet writers may also use the C2A tag libraries in order to publish property changes.) When the Evaluation Navigator portlet process a portlet action (like the user selecting a node in the navigation tree) which should trigger a cooperative portlet event, the action handler will call the following code:

```
  public void publishProperty(PortletRequest
request, String stringToPassThruTheWire, String
type, String name) {

    PortletSettings settings = request.
     getPortletSettings();
    Property p = null;
  PropertyValue[] pva = new PropertyValue[1];

  try {
    PropertyBrokerService pBroker =
     (PropertyBrokerService) getPortletConfig().
     getContext().getService(PropertyBrokerService.
     class);
    if (pBroker != null) {
        p = PropertyFactory.createProperty(settings);
        p.setName( name );
        p.setDirection(Property.OUT);
        p.setType( type );
        p.setNamespace(NAMESPACE);
        pva[0] = PropertyFactory.
createPropertyValue(p, stringToPassThruTheWire);

        pBroker.changedProperties(request,
           portletConfig, pva);
    }
  }
  catch (PropertyBrokerServiceException pbse) {
    CatLogger.debug("PropertyBrokerServiceException:
      nested exception is: " + pbse.getMessage());
    return;
  }
  }
```

Any portlets on the same page that are registered with actions on the same property types (through wsdl files) will receive a regular portlet action, with the action name defined in the wsdl file.

Adding a Third-Party Portlet

So how does a normal person exploit the composite application framework without doing complex coding? This is easily accomplished by using the Portlet Wiring Tool, which is downloadable in WP 5.0 and included with WP 5.1. The Portlet Wiring tool enables to view, add, or delete the current set of wires on a page.

In order to access the Portlet Wiring Tool, click on Edit Page and then click on the Wires Tab. The wires for the page will then be displayed. You can then select the source portlet and output property, the target portlet and input property, and then create a wire.

So now let's go to a IWBCR page and add a third-party portlet.

In Figure 2, we have the Documentation Navigator and the Documentation Detail portlets as delivered by the base product. These portlets are pre-configured (through an XMLAccess script that is executed by the application installer) to "co-operate" with each other through the Cooperative Portlets service.

Again, these portlets use the "programmatic" way of defining and processing the user actions, not the traditional C2A user menu-driven method. When the user clicks on a node in the tree control, the portlet on the right hand side will load the data for the item selected in the tree. Conversely, when the user clicks on an item from the table on the right hand side, the tree will expand to a node that represents that data item.

We can add a third-party portlet (or even multiple portlets) to be deployed on a same portal page with the IWBCR

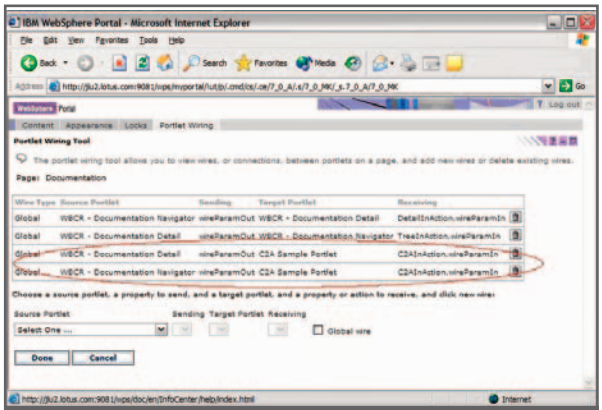


FIG 3: PORTLET WIRING TOOL

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

Improve Web app performance at your pace. Listen to expert Web seminars – it's a fast and easy way to keep up on the latest advice, tips, and best practices.

Visit: www.hwcs.com/wsj19.asp

SILVER ANNIVERSARY 1979 - 2004

H&W

www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap For 25 Years

portlets that can 'cooperate' with the other portlets on the page. The third-party portlet that we will be adding is called the "the C2 Sample Portlet." It has been designed to display a simple message based on the property values emitted by the Documentation Navigator or Documentation Details portlet.

Figure 3 shows the Portlet Wiring Tool screen with the "C2A Sample Portlet," deployed on a portal page with the pre-existing, pre-configured WBCR portlets. This sample portlet reacts to user actions when clicking on the tree nodes, or the items in the table on the right hand side.

As a result of some user action, the IWBCR portlets publish a pre-formatted string message that contains information such as the item type, and most importantly, the item ID for the item selected. Hence, the third-party portlet can use that information to process what it wants to accomplish, in the case of our sample, it simply displays the message. In this fashion, the third-party portlet can work with the WBCR portlets seamlessly. (The third-party portlet can also publish the message to the IWBCR portlets, although we haven't shown that capability in this example.)

```
        <param-value>/wsdl/C2ASample.wsdl
          </param-value>
    </config-param>
```

In the WSDL file that should be packaged with the Sample portlet, the <*types*> definition has to have a simpleType defined as: "SOXNodeWire." (Note that this sample does not need to implement *EventPhaseListener* because the actions are delivered directly to the "actionPerformed" method.)

This matches up with the same type that the WBCR portlets are using so that they can find each other. You can know what types to define for your portlet by inspecting the WSDL files of other portlets. In fact, you can easily change the type names of any portlet simply by editing the WSDL file of that portlet. In this way, an installation could adapt the type names produced and consumed by all of their portlets to be types meaningful to the particular enterprise.

The type definition for SOXNodeWire looks like this:

"Creating and enhancing composite applications with co-operative portlets is not difficult!"

Integrating the Sample Portlet

In order to integrate the sample portlet, we first have to make it a co-operative portlet. When using co-operative portlets, the main servlet class for the portlet is replaced by an IBM specified "wrapper" class. This wrapper class, in turn, takes the original portlet class as a parameter. The wrapper pre-processes many of the co-operative portlet methods and events, then delegates most of its behavior to the original portlet. We define the main servlet class and specify the c2a-application-portlet-class in the web.xml file as follows:

```
<servlet-class>com.ibm.wps.pb.wrapper.
   PortletWrapper</servlet-class>
  <init-param>
<param-name>c2a-application-portlet-class
   </param-name>
<param-value>com.ibm.lwp.c2a.portlet.
   C2ASamplePortlet</param-value>
  </init-param>
```

Then we provide a wsdl file and specify it in the portlet.xml file.

```
<config-param>
      <param-name>c2a-action-descriptor
        </param-name>
```

```
<types>
 <xsd:simpleType name="SOXNodeWire">
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
 </xsd:simpleType>
</types>
```

In your WSDL file you will also define the names of actions that your portlet is willing to perform on the data type you just defined. In our example, the snippet below maps a portlet action name to a WSDL operation on a particular data type.

```
...
<operation name="C2AInAction">
 <portlet:action name="C2AInAction" type="simple"
    caption="any string"
  description="Display an action name"/>
 <input>
   <portlet:param name="C2AInActionMessage"
      partname="Message" caption="message"/>
 </input>
</operation>
...
```

The heavy lifting is in the method: actionPerformed()

"The wiring of a co-operative portlet into a page, and thus into an composite application, is something that can be done by any end-user"

in the C2AsamplePortlet.java. This is where and how the Sample portlet processes the wired message from the WBCR portlets. Remember, the action name and the message parameter are defined in the C2Asample.wsdl file as shown above.

```
public void actionPerformed(ActionEvent event) {
  PortletRequest request = event.getRequest();
  PortletSession psession = request.getPortletSes-
sion( false );
  try {
    String actionName = event.getActionString();
    System.out.println("C2ASamplePortlet:
    actionPerformed: actionName = " +actionName );
    Enumeration requestParams = request.
     getParameterNames();
          while (requestParams.hasMoreElements()) {
       String param = (String) requestParams.
        nextElement();
       String val = (String) request.
        getParameter(param);
       System.out.println(" --param = " + param +
        ", value = " + val);
  }
    // cooperative actions are defined in the WSDL
      file, including the action name
    // and its parameter name. See: C2ASample.wsdl
    if ( actionName != null && (actionName.equalsIgn
      oreCase("C2AInAction") ) ) {
       String wireMessage = request.getParameter
        ("wireParamIn");
       //save the wire message so that it can be
         displayed in the view JSP
  if ( wireMessage != null )
         psession.setAttribute("WireMessage",
          wireMessage );
    }
```

Once the Sample portlet is installed and deployed on a WBCR portal page, one needs to use the Portlet Wiring tool to "wire" the portlets together. Figure 3 indicates what those wires are. Note that the first two wires are the predefined when installing IWBCR application. The last two are the two additional wires to indicate how the Sample portlet re-acts to the IWBCR actions.

The coding above is done by engineers skilled in writing portlets. However, the wiring of a co-operative portlet into a page, and thus into an composite application, is something that can be done by any end-user. The user is not aware of any of the implementation details of properties or actions, but instead simply sees the names of properties and actions that can be interconnected using the portlet wiring tool. This is a small step up from the simple administration required to place a portlet on a page. The task of writing co-operative portlets is also made easier by tooling provided by Rational Application Developer release 6.

This is Only the Beginning

In this article, we showed how to make composite programs with co-operative portlets, both programmatically and with the Portlet Wiring Tool. We showed how easy it was to add a third-party portlet onto a composite portlet. We also showed how IBM was committed to this technology.

But there is only so much you can say in a short article. As we mentioned, we've chosen to show you the more difficult (and more flexible) of the mechanisms for creating composite applications with cooperative portlets, portlet wiring. We did not show you how Click-to-Action portlets can just as easily be integrated as wired cooperative portlets. So if you want to find out more about cooperative portlets, we suggest you read *IBM Developerworks; Using Cooperative Portlets in WebSphere Portal V5* by Amber Roy-Chowdry.

In the last article of our-three part series, we will focus on advanced topics for composite applications. In any programming model, there are a set of patterns that recur repeatedly. For composite applications these include performing portal navigation from components, using labels, using portlets as pop-up windows or "dialogs," enabling "Solo Mode," and forming URL commands to open new windows. We will also look at how to use co-operative portlets to create mapping filters between differing data types. Finally, one of the most exciting capabilities of the composite application programming model is the ability to create associations and user navigational linkages between objects in one business domain and those in another.

The authors wish to thank Joseph Lu, Theresa Smit, and Binh Q Nguyen for their assistance with this article.

Simplifying Development with Eclipse

Take the drudgery out of it!

BY SETH ROBY

Seth Roby has spent his time at Parasoft training representatives from Fortune 500 companies in the Automated Error Prevention methodology. In his role as a Professional Services Engineer, he has written white papers about the process of testing, the infrastructure necessary to ensure quality code, and many other subjects. He has also written and taught courses on Unit Testing practices, with a focus on Java and JUnit. He lives with his wife Sonja in Costa Mesa, CA.

Every IDE will allow you to manage files and build projects. Eclipse goes beyond that by dealing with the code you are writing on a more intimate level than the typical file-centric view embraced by most IDEs. As a developer, you can use that familiarity to your advantage by letting Eclipse do the drudgework of finding, changing, switching, moving, waiting, and just a little bit of writing.

With Eclipse on your tool belt, you can focus yourself on the real task at hand: writing code that does what it's supposed to.

Start with the UI

Let's first note this article refers to the new Eclipse 3, which differs in some respects from older 2.1 builds, as well as WSAD and RAD.

Eclipse looks very similar to most other IDEs because the basic user interface has become so familiar to developers everywhere--except those who still hack away with Emacs or vi, although they too will find something to love in Eclipse, which can support either of those keybindings.

The basic Eclipse UI is shown in Figure 1. On the left of the screen, you'll see a list of development Resources: source code, jars, and other ingredients used to bake your latest development project. In fact, the list is separated into the different Projects that you are currently working on, which can depend on each other or be completely separate.

Eclipse keeps data about the projects in two files, called *.project* and *.classpath*. The *.project* file keeps information about what files are included in the project, how to use them, how to build the project, and many more details. The *.classpath* file can be found in any Java project, and tells Eclipse what to use as the CLASSPATH variable while building.

All of the current Projects exist in your current Workspace. A Workspace in Eclipse is a folder that contains lots of files, possibly including the Resources necessary for the Project. Eclipse starts with a default Workspace; however, you can easily set it to open multiple Workspaces. To open a different Workspace, just use the -data flag when you launch the executable; for example:

```
eclipse.exe -data E:/shared/workspaces/artemis
```

In the middle is a source viewer, which will display the source code for the currently-open file. Notice that it is tabbed, so you can open many files at once. On the right is an outline of the currently-open file. Eclipse lists the classes, methods, and other parts of the file. Double-clicking an item will take you to it. Finally, on the bottom is another tabbed area that is used to convey information about what Eclipse has done, found, or has problems with. Note that each of these lower tabs are Views,

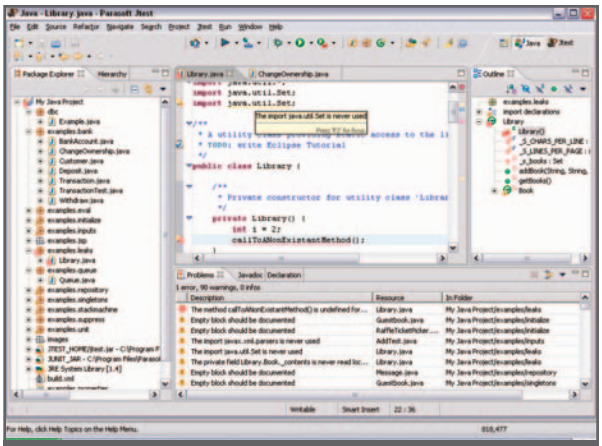


FIG 1: THE ECLIPSE UI

which can be moved around as you like You can combine views into stacks by dragging them on top of each other. The information Views are all separate and can be pulled apart if you have a gigantic screen and want to see all of them at once.

Views are grouped into Perspectives, which are generally environment-oriented; there is a Java Perspective for working with Java projects, a Debug Perspective for tracking down runtime problems, a CVS Perspective for managing CVS connections, and many more. In fact, the number of Perspectives and Views is extensible, as is most every part of Eclipse. The IDE offers numerous ways to plug in and augment the existing functionality, which is taken advantage of by many in the Eclipse community.

Eclipse has Properties for all its Resources as well as its Projects. If you right-click on anything in the list at left, you can select "Properties" to view more detailed information. For files, you can see some basic information about the file itself, like modification dates, location, and permissions. For projects, more advanced options are available that allow you to set up the way that Eclipse will build that project.

If you click on "Java Build Path," you will see four tabs to control build behavior. The first is "Source," which lets you set up where Eclipse will look for source files, and specify which source files to use and which to ignore. The next tab is "Projects," which allows you to set what projects the current project depends on, so that you can set up some libraries or common code in one project and simply include it from other projects.

Speaking of "Libraries," the next tab is named just that and allows you to put JAR files, Libraries, and Class Folders onto the classpath. Finally, you can specify what resources are exported by this project, so that this project's properties will be applied to other projects that import this project.

Eclipse is built around the large source view in the middle, and that's where all of the editing occurs. But note a few things that Eclipse provides aside from the simple text view. First, there is a small trough on the left of the text that Eclipse will use to stash icons that communicate interesting information. A red "x" appears next to lines that Eclipse can't compile. A yellow "warning" street sign appears next to code when Eclipse finds something it thinks is probably wrong, like an unused import statement (as shown in Figure 1).

Second, the code in question can be underlined with an appropriate color. Again, red underlining indicates a compilation error, and yellow indicates a warning. Finally, to the right of the scrollbar, there is another small trough. Once again, colors are used to convey information: red blocks mark code areas with errors, and yellow blocks mark code areas with warnings. Clicking on a block will move the cursor directly to the code in question. Hovering over any of these three color-coded items (icons, underlines, or blocks—all of which are shown in Figure 1) will open a description of the related problem. Right-clicking will bring up a contextual menu.

Automatic Builds

One of the first things you will notice about using Eclipse for Java development is how smart it is about compiling. Go into the preferences (Window Menu > Preferences) and enable automatic building (Workbench > Build Automatically). Now, whenever you save a file, Eclipse will silently build that file and those that depend upon it. As a result, the next time you want to run your application, most of it is already saved and up-to-date. This can be a real timesaver when you are working on a large code base. The simplicity of this feature makes it one of the first things you'll miss if you're forced back onto some other IDE.

"The underlying principal here is that the IDE understands the code"

The underlying principal here is one that will surface over and over as you use Eclipse: *the IDE understands the code.* Eclipse is not just a compiler that can deal with the code and everything else on a file-centric basis. Rather, Eclipse has the smarts to look at the source and deal with it intelligently. Its smart compilation capability demonstrates that intelligence by its understanding of the dependencies expressed in the import statements.

Smart Searching

Eclipse has a very smart Search feature (Search Menu > Search), which allows you to search for occurrences of any string within a file, the entire project, or arbitrary groups of files. That's great, but it gets better. The Java Search (Search Menu > Java) allows you to limit the search to only find classes, methods, types, or other Java-savvy categories. Eclipse can find all of the method implementations that start with "set," and you can easily find all of your write accessors. Better yet, you can control-click on any method invocation to go to its implementation. Or control-click a class name to go to its implementation. To move the other way, you can right-click a method and find all places where it is used (References > Project). Eclipse's understanding of the code helps it navigate a large code base, and you reap the benefit.

"Eclipse offers a host of features that make a developer's life easier by taking some of the drudgework and automating it"

"TODO" Task Link Creation

Eclipse can also remind you about tasks you still need to perform. Many developers add TODO comments into their code when they need to remember a specific task that has yet to be completed, or a certain case that needs to be handled. Eclipse once again draws on its knowledge of Java structure to find any comment that starts a line with TODO and marks it into a special View called Tasks (see it by opening Window Menu > Show View > Other > Basic > Tasks). In this way, Eclipse can provide a comprehensive list of all the tasks that still need to be performed.

Moreover, you can configure Eclipse to recognize other tags that you define. Open the Preferences (Window Menu > Preferences) and browse to the "Task Tags" page (Java > Task Tags). You can add new tags, remove the existing ones, set priorities for different tags, and specify a default for code that Eclipse writes for you.

Code Writing

You'll find that Eclipse can actually write code for you. Right-click a source file and look in the "Source" submenu. These are just a few of the ways that Eclipse can modify or extend your source code, taking a little of the boring work on itself to save you the effort. One of the most useful code writing capabilities is the "Generate Getters and Setters" option, which will scan a class for fields and generate appropriate accessor methods for those fields.

If you select it, a window will pop up to allow you to specify which fields you want accessors for, and which accessors you want (get, set, or both). You can also specify what access the generated methods will have (public, private, etc). Similarly, Eclipse can generate constructors based on the superclass, as well as methods to override or encapsulate superclass methods.

Refactoring

One of Eclipse's most powerful features is its robust support for refactoring. Highlight a method and right-click it, then look at the "Refactor" submenu. If you use this method to change a method name, Eclipse will locate all invocations of this method and change those invocations to use the new method name. You can also change the method signature by adding arguments and having Eclipse change all the existing calls to add that argument, using whatever value you specify. Additionally, you can remove arguments you've decided are unnecessary, or simply rearrange arguments to make the interface more consistent. Refactoring can also work on the class level, where you can pull internal classes into their own files, or move classes to other packages.

Robust Undo Support

Another wonderful feature is called the "Local History," which is a sort of super-simple local CVS repository. Every time you save a file, Eclipse will remember that save and allow you to revert back to it if needed. It remembers saves for the past seven days by default (this can be changed in Window Menu > Preferences > Workbench > Local History).

This can be a lifesaver when you tackle some new feature and end up breaking an unrelated feature. You can do a quick diff between the two versions and see what changes you've made, and you can undo your changes back to the previous version, or the version from yesterday. Better yet, you can revert only certain methods, and keep the changes in other methods.

JUnit Integration

Eclipse is also easy to use with the JUnit framework. If you have classes written to the JUnit standard, you can select the class in the Resources view at left and run the tests (Run Menu > Run As > JUnit Test). Eclipse will automatically find the defined test suite and run all the tests in it, using its own custom UI to display the tests run and the problems found. The familiar JUnit green line turns red if any of the tests fail, and Eclipse will show you the stack trace that generated the failure.

Let Eclipse Do It

So as you can see, Eclipse offers a host of features that make a developer's life easier by taking some of the drudgework and automating it. And because of the platform's architecture, there's a good chance that you can find a plugin that will help you with your specific task. If you can't find it, you can always use Eclipse to write it!

OSBC | OPEN SOURCE BUSINESS CONFERENCE

APRIL 5-6, 2005

THE WESTIN ST. FRANCIS, SAN FRANCISCO

osbc.com | THE BUSINESS OF OPEN SOURCE

osbc.com
THE BUSINESS OF OPEN SOURCE

OSBC is the only conference that focuses on the business of open source strategies and solutions and what they mean to your company. This April, join top level executives and industry luminaries who are paving the way in the Open Source arena.

LEARN to build sustainable, higher-margin Open Source businesses.

HEAR directly from business leaders about how to start and sustain a business on Open Source technology.

NETWORK with your peers and the be first to meet the hottest up and coming companies in the Open Source industry.

CORNERSTONE SPONSOR

Sun microsystems

CORPORATE SPONSORS

intel. Microsoft Novell. ca Computer Associates SDL REALM optaros

Interview with Dick Williams

Wily Technology's CEO talks to WJ

DICK WILLIAMS, WILY TECHNOLOGY, CEO

BY JACK MARTIN

Jack Martin 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, as well as the world's first diagnostic-quality ultrasound broadcast system. Jack is coauthor of ***Understanding WebSphere***, from Prentice Hall. jack@sys-con.com

Wily Technology (www.wilytech.com) provides Enterprise Application Management solutions. The company's products are designed to enable companies to successfully manage their critical Web applications and infrastructure by providing real-time, end-to-end visibility into the performance and availability of these systems. Wily Technology is based in Brisbane, California, just south of San Francisco. WJ's former Editor-in-Chief Jack Martin had a chance recently to sit down with company CEO Dick Williams.

WebSphere Journal: Give our readers a quick overview of what Wily Technologies is all about, if you could.

Dick Williams: Wily allows customers to develop and deploy major Web applications more rapidly and assuredly, and to keep them performing optimally.

WJ: Sounds simple enough, yet your strategy inside of the WebSphere space is unique compared to the other vendors. Could you explain that?

DW: Our strategy is really focused on providing unique functionality to WebSphere customers that allows them to deploy major business-critical WebSphere-based applications reliably, predictably, and rapidly. They could be application server-based; they could be written directly to a JVM; they could be portal applications; they could be integration applications; they could be end-user facing applications; or they could be infrastructure. In all cases we want to allow them to predictably and rapidly deploy those applications and then to keep them in production performing optimally.

WJ: You also have an interesting story in that your products are 100 percent pure Java. Why did you do that?

DW: This is a key part of our current core architecture and technology in that we operate today entirely in the Web space, and largely in the J2EE or Java space. Java actually has matured to do all of the things it originally was designed to do, and enterprises trust it as the core for their complex, business-critical applications. So when you create a complete native Java application, then you inherit an enormous number of capabilities and functionalities. One of the most important is platform support and homogeneity across those platforms. A result of that is we can do a single implementation of a product or functionality, and then immediately the customer is able to deploy it on all Java supported application servers, JVMs and all Java supported platforms. Same identical product, same day, same identical functionality.

WJ: How do you accomplish that?

DW: We do that by adhering to our core architectures and by adhering to the J2EE Java standards. You know that your typical application software team will spend 20 to 30 percent of their time, resources and money on ports—developing, testing and documenting ports. We spend about two percent. Subsequently, we're able to devote that other 18 to 28 percent advancing the product and better serving our customers.

WJ: So the money you are saving by being pure Java you are rolling back into the product?

DW: Absolutely.

WJ: So that is an enormous benefit. We spoke to Barclays Bank in England about a week ago—who is a happy user of yours—and they tried everybody and felt Wily Introscope was the easiest product to deploy right across Barclays Bank. Why is that? (Note the related interview with Barclays Bank accompanying this article.)

DW: A great benefit that you and I have, Jack, is that we both have a long history in the enterprise marketplace, and we

understand just how critical and how complex these environments are. We understand the level of business expectations of these systems and the business impacts that even a small glitch can have. So it starts with a core understanding of that environment, and a commitment to support customers at a business-critical level, as they need. Not necessarily as they expect, because sadly they've grown to have a relatively low level of expectations, but really to support them as they need.

WJ: And Barclays knew what they needed?

DW: When I first met Barclays' John Long, I was very impressed with him. He really understood the totality and the complexity of what Barclays was doing today and what they were trying to do. And he laid out for me a schematic of their entire system, and the way in which he was manageing that system. He had this white space in the middle of the chart and he said, "One big problem is that I can't see inside that white space. I can't see inside that application, I can't see inside that application server or anything associated with it, and it seems to me that that's the most critical part." He really had 'gotten it' long before our people called on him.

WJ: And yet he didn't want to add complexity to be able to see inside that white space...

DW: Right. With Barclays, as with other large enterprises, they try to standardize on as small a number of technologies, products and vendors as they possibly can because as you add additional vendors and products, it adds complexity. Further, once you get to be the size of a Barclays bank you no longer can assume you're going to have a homogenous environment.

WJ: Right, it's next to impossible.

DW: By definition it's heterogeneous no matter how hard they try. And that's one of the great benefits that we deliver to customers. We support that heterogeneous environment as though it was a homogenous environment. So our customers can move an application from one application server to another, from one JVM to another, from one platform to another, from the mainframe to distributed, distributed to mainframe, you name it, and their management is identical. Nothing changes in the way that they manage or support those applications. We provide the same level of insight and provide the same 24/7 product monitoring and management of those applications. We supply the same level of insight and information across the full application life cycle. It goes back to an understanding of the environment in which these enterprises exist—the realities of managing in a very large enterprise environment—and then managing that.

WJ: If you were a customer looking at Wily, what would be the three most important strengths that you would see that Wily's product family has? *DW:* Number one would be customer success. Universally, our customers would tell you that the number one thing that they get from Wily is our fanatical commitment to customer success—no matter what it takes on our part. Number two, I'd say reliability, and number three I'd say responsiveness. Our customers tell me that "you guys do what you say you're going to do. You deliver what you claim, you support it, and when we come back to you with a requirement or a need or a support issue you're immediately responsive to it."

JACK MARTIN SPEAKS WITH DICK WILLIAMS ON SYS-CON TV (WWW.SYS-CON/TV).

WJ: If I were a customer of yours and I got in trouble, what would I expect to have happen?

DW: You'd expect an immediate response from us. If we could not solve the problem remotely, we would be in the customer's facility at the earliest opportunity. And, if the problem involved another company's technology, we would marshal our partner network and stick with the project until the customer was satisfied. In short, you'd expect the best from us, and we would do everything we could to live up to that expectation.

WJ: But a lot of companies have tried in this space; you've actually succeeded.

DW: I think that, number one, that absolute commitment to customer success is why Wily has succeeded. Everybody talks about customer satisfaction. But the reality is that for most companies it's just a lot of words. I believe that if you survey the customers that have implemented any form of Web application management solution, you would find that there is a significant maturity of those customers that are implemented and deployed with Wily, and that they are, in fact, achieving their objectives. That's our commitment. To insure we are achieving that, on a quarterly basis, a vice president from Wily will call on 20 percent of our install base just to find out how they're doing.

WJ: Great, and...

DW: This quarter we will put in place an executive, reporting directly to me, responsible solely for customer success. We're not trying to just measure the level of satisfaction and success…

An Interview with John Long, Barclays Bank

BY JACK MARTIN

Barclays Bank is one of Wily Technology's enterprise customers. Former WebSphere Journal Editor-in-Chief Jack Martin interviewed Barclays' John Long recently.

WebSphere Journal: John, you are head of architectures at Barclays Bank, one of the largest banks in the UK. What does the head of architectures do on a daily basis?
John Long: I'm responsible for setting the technical direction for how we use our technology products to deliver Barclays' business.

WJ: Across the entire bank?
JL: Predominantly the UK bank in application terms and other areas of the group for infrastructure.

WJ: How big is Barclays bank?
JL: Its market capitalization is about 40 billion pounds sterling, and we have around 2,200 branches.

WJ: So why did you and Barclays ultimately choose Wily Technology?
JL: It was two things that came together really. One is that we identified the need for something that Wily does. We were re-engineering and re-deploying all of the systems that support our channels, so we have a lot of different systems supporting the branches, the contact centers, the Internet. We were trying to rationalize those systems with a new set of Java-based applications. And do it in a standardized, multichannel way that would allow us to deploy those systems with a high level of application reuse, such that we could develop them as quickly as possible and make them as effective as possible.

WJ: Why were you focused so keenly on Java?
JL: Java gave us the best mix of portability, capability and also high levels of security and manageability.

WJ: What did you learn in deployment?
JL: We found that because we were deploying all of our applications to all of our channels from a mid-tier platform that if we did have a problem it affected everything. So the impact of failure, although failures were quite rare, was huge. What we were finding was because they were a new set of applications, although we were monitoring and managing the infrastructure very well–the application server, the database, the security and the network–we still couldn't find where the problems were.

WJ: And this is where Wily came into the picture.
JL: Yes, we bought and integrated a lot of software into our environment and Wily is one of the simplest and one of the most straightforward to integrate. And it also has one of the least actual operational impacts. It doesn't consume a lot of your time or resource to actually get it live.

WJ: Why is that?
JL: I think it comes from a really, really great idea in the middle. The way it goes about things, that isn't a tack-on, added-on, "bring later on" approach. It actually goes to the core of Java applications.

WJ: How many people do you have using Wily's product right now?
JL: In development, all our key Java applications would go through it. Enterprise-wide. We standardize our testing so it goes through a standard model of people. So there are probably 50 to 60 people who test with it, and in deployment there are probably about 20 people who use it.

WJ: So everybody is using it. When you chose Wily did you have a bake-off where you looked at alternatives?
JL: We did.

WJ: So they won fair and square?
JL: They did.

WJ: So you didn't just pick out Wily because you liked them; you picked them because they had a great product.
JL: Yes, they also did well because the management teams were particularly sharp. They had to hurdle the fact that they weren't IBM.

WJ: If you were to advise someone looking at Wily, what would be the number one reason that you would say that they should consider using Wily's product?
JL: The first is recognizing the importance of being able to do application management as well as infrastructure management. If you want to deploy an application management tool, which, if you are delivering on an enterprise scale, I think you should, then I think its ease of introduction, the fact that within a small number of months it will be live in your operation, and also the low level of impact it has on your environment. We've seen other management tools take a lot of resource to collect the kind of information that Wily collects without needing that high level of resource.

WJ: And where do you see yourself going with Wily over the course of the next 12 to 36 months?
JL: I think the main area will be to continue to use it across our application space, so as we deliver new applications, we will implement those using Wily. But I think one of the main things that we want to do is start giving it the business focus to making our senior business leaders have business-type or business-facing Wily dashboards so that they can see the live operation of the applications that run their business.

WJ: You'll be more pro-active than that.

DW: What I expect is that we'll seek out problems, issues, opportunities and then actually do something about them. Our number one objective is customer success and everything that entails.

WJ: Within that context, you could have worked with a lot of different companies. Why did you choose WebSphere?

DW: Lewis Cirne, our founder and Chief Technology Officer, originally conceived of this technology. He had the good fortune to call on IBM, and IBM helped focus his attention on the server as opposed to the client. This goes back about six years ago. And so we began working with IBM very early on. Now, we also work with BEA, SAP, Oracle, Sun Microsystems, JBOSS, you name it.

WJ: I know you work with everybody, but isn't WebSphere the biggest pocket?

DW: We've had a very special relationship there for a very long period of time and it gets back to our focus on customer satisfaction. We do have superior technology and we have been able to deliver some very innovative solutions to the marketplace. Another key part of it has been understanding IBM and understanding their customers—their unique demands and requirements and the ways in which IBM will support them. That means future products and enhancements and everything else. Having spent 22 years at IBM myself, I have a good understanding of the people and culture. At Wily, we spend a lot of time working with IBM

"Most of those performance impacts are caused by, guess what, change"

to understand the company's advanced ideas and strategic plans. That way we can put into place the core technologies and infrastructures on our side that will support them when they come to market.

WJ: IBM traditionally has extraordinarily demanding customers. You're thriving inside of that environ-

WebSphere Journal | Advertiser Index...

COMPANY	URL	PHONE	PG
BLOG-N-PLAY.COM	WWW.BLOG-N-PLAY.COM	201-802-3000	21
EV1 SERVERS	WWW.EV1SERVERS.NET	800-504-SURF	15
H&W COMPUTER SYSTEMS	WWW.HWCS.COM	800-338-6692	35
IBM	WWW.IBM.COM/MIDDLEWARE/FLEXIBLE	800-426-4968	11
INFERDATA	WWW.INFERDATA.COM/WDJ	888-211-3421	39
INSTANTIATIONS	WWW.INSTANTIATIONS.COM/ANALYTIX	800-808-3737	17
ITKO	WWW.ITKO.COM	877-BUY-ITKO	51
IT SOLUTIONS GUIDE	WWW.SYS-CON.COM/ITSG	201-802-3021	43
JINFONET	WWW.JINFONET.COM	301-838-5560	29
MOTIVE	WWW.MOTIVE.COM/WITHIN1	512-531-2527	52
PARASOFT	WWW.PARASOFT.COM/PRODUCTS	888-305-0041	2
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	3
QUESTSOFTWARE	WWW.QUEST.COM/WSDJ	949-754-8633	25
SYS-CON MEDIA REPRINTS	WWW.SYS-CON.COM	201-802-3026	13
SYS-CON MEDIA E-NEWSLETTERS	WWW.SYS-CON.COM	201-802-3020	47
TANGOSOL	WWW.TANGOSOL.COM	617-623-5782	37
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5

WebSphere Journal | Coming Next Month...

The Support of Composite Applications in WebSphere Portal *Part 3*

How to Add Logging to Portlets

The Value of Modeling in Illuminating WebSphere

High Performance Server-Side SOAP Performance

ment, and I guess part of it is that you come from that environment.

DW: Yes, we love demanding customers. They're great! These customers' systems are no longer merely important. They are truly business-critical. When you have a technology that is fundamentally an integrative technology, one that ties everything in your business together, when it starts failing or when it starts performing poorly, it's not just a business inconvenience; it's a major business impact. To support these customers, you've got to be willing to deal with both your partners and your customers with a level of responsibility that demonstrates real respect for their business.

WJ: Dick, where do you see the market going in the next year or two?

DW: I think that application management will become more and more important. Today it's all about Java. Increasingly, it will be all about an integration of a broad variety of platforms and technologies, including Java, (Microsoft) .NET, and a whole lot of different things. In the future it is about integration of all of the things that the customer has historically done in different ways, and allowing them to manage all of that in a wholly new way. That new way, of course, is centered on the business significance of the application, and how effectively that application is delivering value to the business. In the future Web application management will be all about providing a more complete and broadly-based collaborative management solution to the totality of what enterprises are doing in this new environment.

WJ: Security is becoming a really big issue. When you are looking at applications on a micro-level, what kind of implications does that have for security right now?

DW: The implications that it has is that over time we ought to be able to detect abnormal and unanticipated behavior. The first level of that, of course, is change, and the reality of these new environments is that change in endemic. It's a part of the process. It's a part of the reason why they're implementing these new technologies, because the business requires IT to be far more responsive to change and to be able to implement new ways of doing business on a continuing and very rapid basis. As you do that, of course, it totally upends the ways in which you develop, test, deploy and support applications. You just cannot test enough quality into these applications. You can't test your way through an unanticipated change that's going to happen at three o'clock this afternoon, or 12 unanticipated changes that just happen because of the loads that you have on the system, not because of anything done to the system. A lot of those changes could well have to do with security. So one of the things that we are beginning to look at is with this deep level of knowledge and real-time information, what else can we do that would allow customers to manage proactively, based on the ability to predict changes or their outcomes.

WJ: So what do you think is the Achilles heel of the typical IT infrastructure today?

DW: It's probably change. Historically, Jack, when we developed mainframe systems or client-server systems, we tested the heck out of them, we integrated them, we threw all kinds of load at them–and then we deployed them. We'd watch them for a little while and then we'd lock them down and we'd allow no change except on a yearly or a quarterly basis at most. But the reality is that businesses today are changing so rapidly and they have such a tremendous need for agility that they need IT to be able to respond very rapidly. The key reason they've gone to Java and other web-based technologies is because they allow them to achieve that level of agility and responsiveness. In addition, businesses are saying they want more value from their enormous investment in CRM, ERP, personnel systems, etc. They want to be able to tie these systems together. Novel idea! They want to provide access to employees, suppliers, partners and customers to all of that information and all those systems, and, of course, these Web technologies allow them to do that. It gets to be very complex but it allows them to achieve the level of agility the business needs. And then, they realize everything is interrelated and a problem or single change in one area can have broad reaching impacts across the entire system.

WJ: Yet change is always with us.

DW: Yes, change is endemic, so I think it is the real Achilles heel. I think it's far more impactful than security *per se.* Change opens up holes that can allow security breaches, and it also can have devastating impacts on performance on both the IT and business levels. Our IT systems, our business environment and our performance expectations demand that we adopt an approach to application management that accepts and embraces change.

"Everybody talks about customer satisfaction. But the reality is that for most companies it's just a lot of words"

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

Offering leading software, services, and hardware vendors an opportunity to ***speak to over 100,000 purchasing decision makers*** *about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management*

Don't Miss Your Opportunity to Be a Part of the Next Issue!

Get Listed as a Top 20* Solutions Provider

For Advertising Details Call 201 802-3021 Today!

*ONLY 20 ADVERTISERS WILL BE DISPLAYED. FIRST COME FIRST SERVE.

SYS-CON MEDIA

The World's Leading i-Technology Publisher

The Year of SOA

BY ROGER STRUKHOFF

Let's take a quick generational placement test. Who among you remembers The Year of the LAN? Sorry, I'll speak up. WHO AMONG YOU REMEMBERS THE YEAR OF THE LAN? Raise your arthritically-challenged arms, please.

The Year of the LAN was this mythical beast in the early to mid-80s that was going to transform the industry by interconnecting all information into one seamlessly interoperable Information Oz. It didn't come the first year that it was predicted. It didn't come the second year it was predicted. After a third year of waiting, everyone went quietly back to work, muttering under their breaths that these computer thingies were never going to work, that they would never be connected in any meaningful way, and that media/analyst predictions fell somewhere between a late 70s-model American car and dousing on the predictability scale.

But then it happened. Once everyone stopped hyping it, it turned out that local-area networks became ubiquitous in the office environment and not uncommon in homes. Wider-area networks of all bandwidths and topologies also proliferated. The development of the Web and popular emergence of the Internet in the 90s sealed the deal. Most users now thought of their computer as a gateway to vast amounts of information, whether corporate or external, rather than just a dumb, future boat anchor that cranked out text and numbers.

"The Year of the LAN was this mythical beast in the early to mid-80s that was going to transform the industry by interconnecting all information into one seamlessly interoperable Information Oz"

The world of information, if not exactly a seamlessly interoperable Oz, became very highly integrated and the concept of the "floppy shuffle" took its rightful place in the recyclable dustbin of IT history.

Yet, industry veterans can still muster a cynical smile when they think of the Year of the LAN and how it "never happened." The problem is, it *did* happen, just not the first year that it was *announced* that it was going to happen. The Year of the LAN is thus remembered as a disappointment to many, when in fact the development and proliferation of effective IT networks is one of the top two or three great developments in the IT business over the past 15 years.

The game of setting up unrealistic expectations only to dash them when a lack of immediate results present themselves is one of the great ploys we media hacks use to sell magazines and newspapers. This game is so enjoyable, it's played by media people in all walks of the journalistic life. For example, I remember reading in 1985 how the young Roger Clemens needed to mature very soon or chances were he was not ever going to realize his potential as a major-league pitcher. How many times have you read articles early in a new Presidential administration how appointments are going so slowly that it was clear that this was going to be a lackluster, even failed, Presidency? And certainly around this time of year, every year, we're bombarded with constant stories wondering if it is ever going to stop snowing and being cold?

Thus we now enter The Year of SOA era. Let's promise ourselves to be patient. Service-oriented architectures are today an arcane subject to most people on the street, and even to many in the IT business. LANs were the same way two decades ago. Yet over the years as the technologies became easier to manage and plug-and-play became more of the norm, even the casual computer user has developed a relative comfort level with their network.

SOA will be the same way some day. Developers will understand the importance of the overall architecture, IT management will become comfortable with the Web-centric approach to application development and lose its fear of exposing too much IT infrastructure and information to the public, and the casual user may even some day understand how the modern-day SOA has made their access to information easier and much more productive.

But let's be careful about claiming any specific year, including this one, as The Year of SOA. We're in this for the long haul. And maybe 20 years from now those of us who are still around can remember how we were smart back at the dawn of the 21st century not to expect too much, too soon.

Roger Strukhoff, editor-in-chief of ***WebSphere Journal***, is West Coast Bureau Chief for the SYS-CON News Desk, and President of www.wdva.com. He spent 15 years with Miller Freeman Publications and The International Data Group (IDG), then co-founded CoverOne Media, a custom publishing agency that he sold in 2004. His work has won awards from the American Business Media, Western Press Association, Illinois Press Association, and the Magazine Publishers' Association. Read his blog at http://www.rssblog.linuxworld.com. Contact him at roger@sys-con.com.

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.

Motive brings an enlightened approach to application management, building management intelligence into the application itself. Only Motive transcends the fragmented, disjointed reality of traditional application management, to deliver the visibility, insight and automation that support and development teams need to keep applications running smoothly.

This is no mere vision for the future. It's here now in application management products from Motive. Learn more about Motive's breakthrough approach in a new white paper about built-in management at www.motive.com/within1.

motive

www.motive.com