STRATEGIES • SCA • EJB • SOA • NEWS

www.WebSphere.SYS-CON.com

WebSphere® JOURNAL

The World's Leading Independent WebSphere Resource

JANUARY 2006 VOLUME 5 ISSUE 1

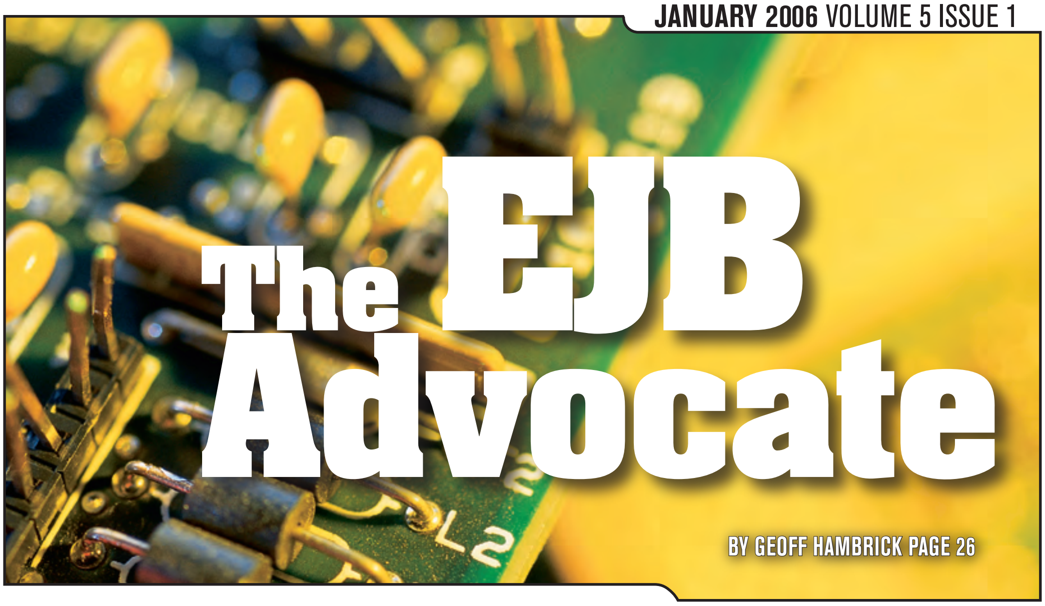
CALL FOR PAPERS NOW OPEN!

SOA 10th International WebServices Edge conference+expo

2006 ENTERPRISE> OPENSOURCE CONFERENCE+EXPO

SEE PAGE 29 FOR DETAILS

Coming... June 2006 New York, NY



The EJB Advocate

BY GEOFF HAMBRICK PAGE 26

PRESORTED
STANDARD
US POSTAGE
PAID
ST. CROIX PRESS

NEW!

CodePro AnalytiX™

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.

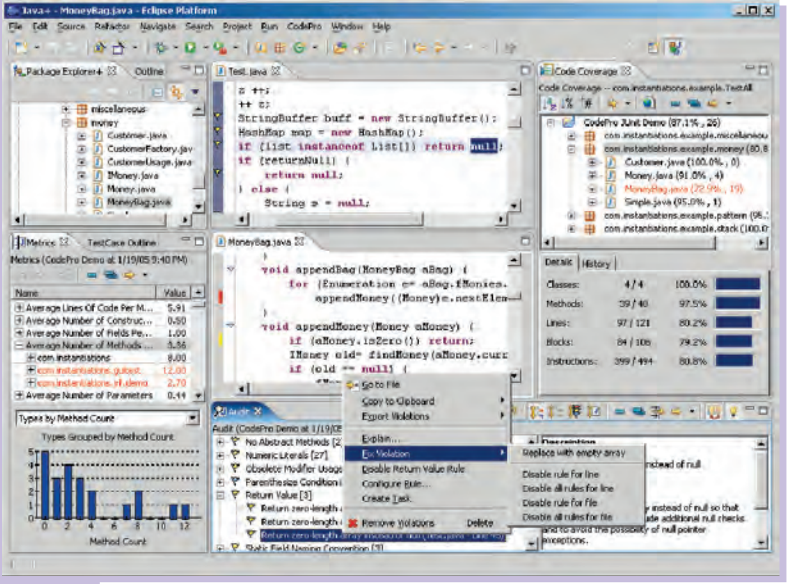
From the development tool experts at Instantiations. No one outside of IBM has longer or deeper experience in creating Eclipse-based Java development tools.

Put our experience to work for your team.

www.instantiations.com
1-800-808-3737

Copyright 2005 Instantiations Inc. CodePro AnalytiX and CodePro are trademarks of Instantiations.
All other trademarks mentioned are the property of their respective owners.

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



```
z ++;
++ z;
StringBuffer buff = new StringBuffer();
HashMap map = new HashMap();
if (list instanceof List[]) return null;
if (returnNull) {
   return null;
} else {
   String s = null;
```

```
   }
   void appendBag(MoneyBag aBag) {
      for (Enumeration e= aBag.fMonies.
         appendMoney((Money)e.nextElem
   }
   void appendMoney(Money aMoney) {
      if (aMoney.isZero()) return;
      IMoney old= findMoney(aMoney.curr
      if (old == null) {
```

Metrics (CodePro Demo at 1/19/05 9:40 PM)

Name	Value
Average Lines Of Code Per M...	5.91
Average Number of Construc...	0.50
Average Number of Fields Pe...	1.00
Average Number of Methods ...	3.36
com.instantiations	8.00
com.instantiations.gutest	12.00
com.instantiations.irl.demo	2.70
Average Number of Parameters	0.44

Code Coverage -- com.instantiations.example.TestAll

CodePro JUnit Demo (87.1% , 26)
com.instantiations.example.miscellaneou
com.instantiations.example.money (80.8
Customer.java (100.0% , 0)
Money.java (91.0% , 4)
MoneyBag.java (72.9% , 19)
Simple.java (95.0% , 1)
com.instantiations.example.pattern (96.:
com.instantiations.example.stack (100.0

Classes:	4 / 4	100.0%
Methods:	39 / 40	97.5%
Lines:	97 / 121	80.2%
Blocks:	84 / 106	79.2%
Instructions:	399 / 494	80.8%

Audit (CodePro Demo at 1/19/05
No Abstract Methods [2
Numeric Literals [27]
Obsolete Modifier Usage
Parenthesize Condition
Return Value [3]
Return zero-length
Return zero-length
Return zero-length array instead of null
Static Field Naming Convention [3]

Description
instead of null
instead of null so that ade additional null checks and to avoid the possibility of null pointer exceptions.

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

Open Up!

Portals–Get what
you want easily

company, product and service names maybe trademarks or service marks of others.

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

FROM THE EDITOR

The Need for Big Apps

BY ROGER STRUKHOFF

It will be interesting over the coming year to witness the progress of open source software in general and open source in application development in particular. IBM, chided by many members of the open source community for taking the proprietary approach to app development (through WebSphere, for example), nonetheless claims to be a strong proponent of open source approaches where applicable.

And it's applicability that may determine the ultimate fate of open source software and of the debate surrounding it. The offices of SYS-CON Media, publisher of this magazine, were treated recently by executives from a leading application integration vendor, who outlined his company's global strategy in addressing the needs of a customer base that has the most extensive demands imaginable.

It seemed clear from discussing things with these execs that there are certain areas, many areas, in which the rich development environment of a WebSphere remains an essential ingredient to mission-critical success. Such things as airline reservations systems, for example, with their high number of transactions, confidential credit card information, and necessity to sync up reservations information with company operations, are something that one can imagine very few companies trusting to anything less han the most full-featured development environment imaginable. Talk from open source boosters of paying proprietary "tax" to vendors such as IBM can seem to be a fallacious argument when one considers the actual needs of the application and the inherent risk in developing it to the needs of the customer.

That said, this issue of *WebSphere Journal* delivers another suite of articles targeted at these demanding environments. Tilak Mitra offers the first of two articles about Service Component Architecture. Advanced Authentication in WAS, a critical feature inherent in the world of the app server, is addressed in one of our features this month. Other prime topics include EJB and another article about Service Component Architecture, this one about building SOA solutions within this context. Another WAS article, this one about adopting non-SOAP HTTP, addresses another important element that developers and architects often face in their mission-critical world. And we also feature the second article in a series focused on overall software development project management success. IBM has developed a two-tier approach to the apparent open source vs. proprietary dichotomy, and this publication continues to cover that upper-tier world, characterized by large companies doing large applications. Other vendors, notably BEA, try to square the circle by offering blended approaches and dividing the application development world into application types throughout all enterprises, rather than looking at typical small-business vs. large-company needs.

So the debate will continue. But the marketing-department positioning by major vendors, including IBM, and the continuing open source debates, which tend to cast people on each side of the argument as either "right" or "wrong," are of little consequence to the forward progress of WebSphere developers. Whether developing an airlines reservation system, a global supply chain for a major industrial or retail company, an information portal and intranet that covers customers, suppliers, and employees, it is clear that there are major parts of the IT world that will continue to demand the most highly developed environment imaginable. To those readers, we continue to offer *WebSphere Journal!*

ABOUT THE AUTHOR

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.

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
ASSOCIATE EDITOR	SETA PAPAZIAN

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	GAIL NAPLES
ACCOUNTS PAYABLE	BETTY WHITE
ADVERTISING DIRECTOR	ROBYN FORMA
ADVERTISING MARKETING DIRECTOR	DENNIS LEAVEY
ADVERTSING SALES MANAGER	MEGAN MUSSA
ADVERTSING SALES MANAGERS	DOROTHY GIL KERRY MEALIA
LEAD DESIGNER	ABRAHAM ADDO
ART DIRECTOR	ALEX BOTERO
ASSOCIATE ART DIRECTORS	ANDREA BODEN TAMI BEATTY LOUIS F. CUFFARI
VIDEO PRODUCTION	FRANK MORICCO
WEB DESIGNERS	STEPHEN KILMURRAY VINCENT SANTAITI SHAWN SLANEY
ONLINE EDITOR	ROGER STRUKHOFF
CIRCULATION SERVICE COORDINATORS	EDNA EARLE RUSSELL LINDA LIPTON
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

© COPYRIGHT 2005 BY SYS-CON PUBLICATIONS, INC. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT WRITTEN PERMISSION. FOR PROMOTIONAL REPRINTS, CONTACT REPRINT COORDINATOR, CARRIEG@SYS-CON.COM. SYS-CON PUBLICATIONS, INC., RESERVES THE RIGHT TO REVISE, REPUBLISH AND AUTHORIZE THE READERS TO USE THE ARTICLES SUBMITTED FOR PUBLICATION.

ALL BRAND AND PRODUCT NAMES USED ON THESE PAGES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES. SYS-CON PUBLICATIONS, INC., IS NOT AFFILIATED WITH THE COMPANIES OR PRODUCTS COVERED IN WEBSPHERE JOURNAL.

WEBSPHERE® IS A REGISTERED TRADEMARK OF IBM CORP. AND IS USED BY SYS-CON MEDIA UNDER LICENSE. WEBSPHERE® JOURNAL IS PUBLISHED INDEPENDENTLY OF IBM CORP., WHICH IS NOT RESPONSIBLE IN ANY WAY FOR ITS CONTENT.

SYS-CON MEDIA

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

©2004 Wily Technology, Inc. The Wily logo is a trademark of Wily Technology, Inc. Java is a trademark of Sun Microsystems in the U.S. and other countries.

A personal perspective

Strategies for Software Development Project Success PART 2

BY GORAN BEGIC

In the first part of this series, I examined two factors that are essential for project success: compensating for lack of face-to-face communication and writing better use cases.

ABOUT THE AUTHOR

Goran Begic is a Senior IT Specialist with IBM.

In the final part of this series, I will outline two additional elements that are vital in software development:

1. Ensuring effective testing; and
2. Supporting marketing efforts.

I. Ensuring effective testing

Once the use cases are set in place and the team has agreed that they represent the right way to go, the use cases become the foundation for the rest of the plan. In fact, this is the only way to take advantage of the benefits they offer.

The engineering team builds a development plan that includes, at the very least, a list of components to be built and a timeframe for each of them. It is very important to create clear traceability between features needed for the main use cases and the components necessary for the features to work.

Identifying these core components and defining their use cases are crucial steps that allow early testing of the application functionality. If the core components are delivered early in the development cycle, then the tester can start writing test scripts for the basic set of rules and validate that the tool functions properly.

In our example, the system use case "Run code review" enabled the tester to make a test plan for this core functionality even before the code was written and also to create a set of manual test scripts for both the main flow and alternative options.

Types of testing

The simplest form of testing – and a very effective one -- is to assemble a number of educated users to exercise various features of the application under test and report issues (findings, defects) to the code development team. The metrics for this form of validation are simple: The more users you have, the more defects you'll detect. Different user groups will use the tool in different ways and further improve the number of detected problems. However, there are some issues related to this. By the time the software is ready for user consumption, there may not be enough time to launch an extensive test program. Different users may be on different product builds. Even more important, depending entirely on human beings is very expensive and very unreliable.

There are still more things to consider. Without a clear testing plan, it is impossible to assume that the same features of each version of the application will be tested the same way. Without assistance from testing tools, the sequence of test steps will most likely differ for each test, and there will be slippage on some rare -- but potentially important and costly -- scenarios.

Manual testing and regression testing

Manual testing refers to a set of actions aimed at validating a specific system response. The alternative to manual testing is automated testing. Both are types of functional testing. Automated testing implies that you use a specialized tool or batch script to exercise a set of application components, record the application's response, compare it against expected values, and decide whether the test was successful or not -- all without any human intervention. Automation gives you the ability to repeat the same tests over and over again, with much better precision than humans would ever be able to achieve. Examples of automated functional testing tools are IBM Rational Robot and IBM Rational Functional Tester.

Regression testing, which is used to measure application quality, can be either

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

© 2004 Motive, Inc. All rights reserved. Motive, the Motive logo, and all other Motive products and technology names mentioned herein are trademarks of Motive, Inc. All other products or services mentioned herein are trademarks of their respective owners.

manual or automated. You can assemble a regression testing suite from automated functional tests or from automated developer tests (see below). The key element for reliable regression testing is exact repeatability for these tests. Therefore, it is necessary to precisely define test steps in documents called test scripts, and then follow these exact steps during each regression test. Then, you can confidently use the test results not only to report problems, but also to measure quality.

There is a high correlation between success in testing and the amount of time you invest in test planning, documenting manual tests, and automation. Here are some specific suggestions for effective testing:

- Define your test plans around use cases. Start testing the main use-case flows first, and then expand into alternative flows once you cover all the main use-case scenarios. The key is maintaining the proper granularity and modularity for your use cases, as described above.
- Organize your manual tests around a test plan, and start documenting and analyzing test results in a uniform way with the first batch of tests that you implement. Repeatable, uniform execution (even if manual) will improve the quality of metrics that you collect over time.
- Automate first the tests that have relatively simple possible execution paths but require a lot of data to be entered with each run. Feed the test scripts with ready-made data pools (i.e., do data-driven testing).

Developer testing

Often, there is a big obstacle to converting use cases to effective tests: A large portion of the code base is not available for functional testing until late in the development cycle. Therefore, it would be good if some components were tested before they were assembled into the running application. This is where developer testing fits in.

Developer testing is a set of activities focused on improving code quality and often conducted by a developer. Developer testing has two main aspects:

- Automated unit and component testing. This includes code reviews, unit/component testing, and code-coverage analysis.
- Manual testing and debugging. This includes execution trace, assertions, memory leak detection and memory usage analysis, performance profiling, thread analysis, and so forth.

Automated batch tests with dedicated tools such as JUnit – a code review tool integrated in IBM Rational Application Developer – or IBM Rational PurifyPlus provide an additional means to ensure high-quality software. Finding and fixing defects in the development environment means fewer functional problems later on. And that leaves more time for writing code and introducing more automation. In addition, having reliable repeatability for these automated unit tests and code reviews allows you to collect valuable metrics about code quality, with the same benefits described previously.

For most organizations, the main hurdle to implementing developer testing is the learning curve for the required tools. Often, individual developer testing tools focus on a rather narrow aspect of software quality. If team members do not have experience with automated developer testing, then finding the right tools and deciding what types of tests to automate can present considerable challenges. Therefore, the development plan should not only build in time for developing and debugging application components, but also dedicate time and resources for the training, setup, analysis, and reporting involved in implementing automated development tests. This initial investment will quickly bring returns by reducing the number of functional problems left for the dedicated testing teams to detect. It will also raise the level of understanding of the code base among team members.

Here are suggestions for getting started with developer testing:

- If you are doing unit testing, start collecting code coverage data while running the unit tests to assess the completeness of the unit testing suite.
- If you are developing C++ applications, run your key use cases with the tool for analysis of dynamic memory allocation. Memory corruption problems are one of the key problem areas in all native C/C++ applications, and the root cause of many unexpected and hard-to-reproduce defects.
- Collect performance baselines for the methods in your components for at least each integration build, and monitor the development of performance over time.
- If you are developing a Java/J2EE application, monitor the memory usage and the number and types of objects in memory during a couple of basic smoke tests.
- Apply a handful of the most important static analysis rules at the beginning, use them against each build, and add more rules for your code base over time. This will reduce the number of false positives in the results.

Developer testing doesn't replace functional regression testing -- manual or automated. It simply improves the effectiveness of testing as a whole, by detecting problems in the development environment, when they're relatively easy to fix.

Positive versus negative testing

Positive tests focus on validating an application's main flows, as defined and prioritized in the use-case documentation. Negative tests often focus on testing the application's capability limits (i.e., on "breaking the application").

In my opinion, a good test plan should clearly focus on validating use-case paths, but include a healthy dose of limit testing. Often, you can assess limits through data-driven testing, which applies different data sets against a single test case in order to validate the application's response to a certain problem or exception – non-standard character sets, for example.

Supporting marketing efforts

In today's business-driven, dynamic environment, understanding the software market is key to project success. In fact, an

TANGOSOL™

A Cure for Downtime.

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

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

Try before you buy.
Download a free evaluation license at
www.tangosol.com

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

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

organization's degree of understanding may have an even bigger impact on success than the level of communication, project planning, quality, and testing that you achieve. Even a perfectly organized, well-tuned software development organization will fail if its product misses the needs of the market or targets the wrong audience.

So how can those who are doing the engineering work with people in marketing to improve a product's chances for success? Below are some suggestions.

Driving beta programs and customer references

The most effective way for software teams to help market a new product is to obtain positive references from early customers and share them with prospects via their organization's marketing channels. No matter how skilled or creative your marketing people are, they can't collect success stories overnight. It takes time for customers to familiarize themselves with the new tool and apply it. Therefore, planning for the new project should include a significant amount of time and resources for an early adoption program.

Begin at the outset of the project by engaging the marketing team, a number of field representatives, and a customer from your target audience in discussions about features and other requirements for the new product. Get in touch with key customers who requested enhancements in your product, present a plan and timeline for a beta program, and make sure that this plan includes the customer reference as a target. By communicating this requirement ahead of time, you can help set correct expectations for the program and improve the chances of meeting your marketing timeline. Often, customers must go through a number of legal hoops and approvals before they can provide a reference that will help you highlight the new solution's benefits.

Early collaboration

It's also a good idea for software engineering to start working with the marketing organization early in a project, because marketing people can help you obtain the information necessary to create effective use cases. They can help you define your target audience and market because they constantly work with focus groups and analysts who maintain a clear picture of the current market. At the same time, the work you do to define goals for a feature or product release will yield well-prepared documentation that can help the marketing organization understand the product early on and enable it to describe and promote the product accurately. The key word is collaboration.

A good practice is to assemble a collaborative project team to help everyone in the organization understand what needs to be done. In addition to gathering the marketing input described above, this collaborative team should also do the following:

- Recognize key sponsors and engage them in the planning phase (review of goals, vision, business goals, and use-case scenarios). The sponsors can be external investors, potential customers, or even groups within your organization.
- Get information about the client's support requirements, including their level of expertise and training needs.
- Get information from the sales team about the desired availability date, key customers, and major concerns.
- Assemble presentation material to introduce other parts of the organization to the project so that they can begin participating. Collaborative team members should stay up to speed on project goals, plans, and activities throughout the project lifecycle so that they can continue to educate interested parties.

"Talking" marketing

"Talking" marketing refers to how you communicate technical features through the media. Sometimes the target audience for a marketing effort is very technical, but more often, such efforts are pitched at business decision makers who are not technical at all. The best way to help your marketing organization transform feature details into meaningful information for a non-technical audience is to fully understand the benefits and value of a new product or enhancement, and be prepared to help express them in metrics such as dollar amounts.

Be ready to provide clear answers to questions such as:

- "Why is this new product or enhancement important?"
- "How does it support the key initiatives in our business strategy?"
- "Who will benefit from the new capabilities?"
- "How does the product work?"

In my experience, there is always a way to explain even the most complex technical solutions in a relatively simple manner that summarizes the motivation behind the solution and its benefits. The challenge is to find a model that most of the people in your audience can understand.

Conclusion

In the face of the complex business and technology demands that globalization has brought about, today's organizations have to concentrate both non-technical and technical resources on creating a successful product and a strong market presence. Although there is no silver bullet that can protect your software development organization from all the challenges it might encounter in the course of a project, recognizing the importance of synchronization among all the different departments and functions within your organization will better prepare you for meeting every challenge that comes your way.

Notes

Standards for writing code can be industry specific and are often mandatory requirements.

Imagine seeing EAI's impact on business process performance in real-time...

Now imagine implementing it in minutes*,* *instead of* months

To get your free copy of the informative new article

"3 Steps to Uncomplicating Business Process Management"

go to www.nastel.com/uncomplicate
or call 800-580-2344 or
+631-761-9100

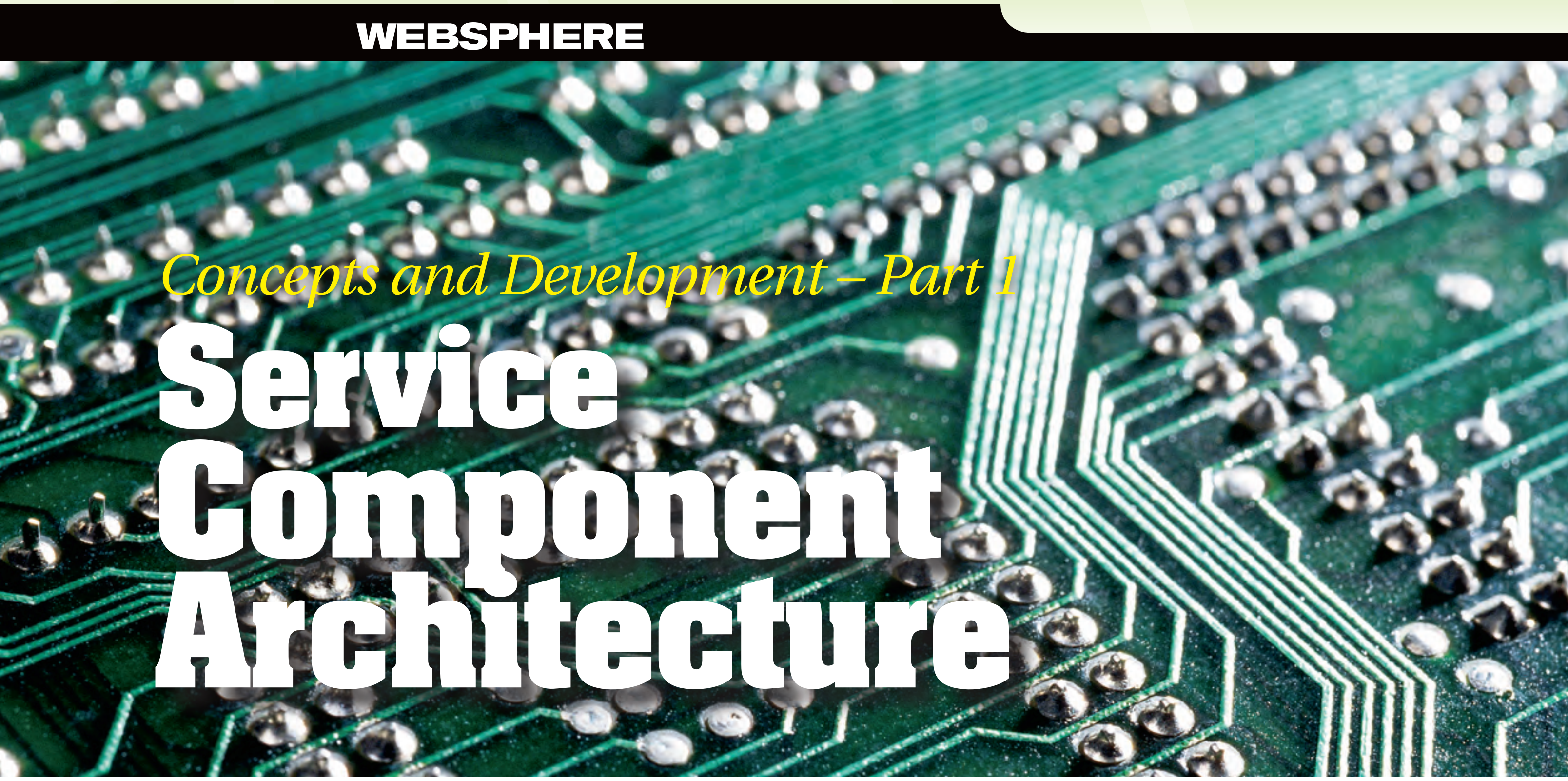
Business happens in real time.
If something impacts your business processes, you need to know. *Now.*

AutoPilot™ from Nastel monitors and manages all of the complex events that comprise your business-critical transactions, giving you deeply granular visibility from end to end for all your business processes. AutoPilot lets you define customized, actionable alerts in the context of each user's specific business function to streamline your business processes and eliminate costly failures, bottlenecks and redundancies. With AutoPilot, you can see the impact of EAI on your business process performance in real time, on-demand.

Unlike other monitoring software, though, AutoPilot's unique architecture takes the complexity out of implementation. It doesn't require months of programmers and consultant's time before you reap the benefits. With AutoPilot, you simply point and click, and within minutes you're in total control.

NASTEL™
TECHNOLOGIES, INC.

Put Your Application Infrastructure On AutoPilot



Concepts and Development – Part 1

Service Component Architecture

BY TILAK MITRA

In the world of IT and computer software development, Service-Oriented Architecture (SOA) has been the architectural find of the 21st century that promises to solve most of the problems that has plagued software development efforts over the previous few decades. The IT industry now considers SOA as the Holy Grail to solve the inherent problems of inflexible, high maintenance software applications.

SOA is an architectural paradigm that is based on the foundational concepts of loosely coupled business oriented services that can be composed and recomposed to realize enterprise business processes.

While SOA creates a service-based architecture, not only do the services need to be specified and developed, they need to be composed to create and realize end-end business processes.

Service Component Architecture (SCA) is a specification which describes a model for building and integrating applications and systems using SOA. SCA aims at simplifying the means to create business applications aided by sophisticated tooling which implements an SCA.

This two part article series is going to provide a broad overview of SCA and how the WebSphere Integration Developer (WID) integrated development environment (IDE) can be used to build and integrate end-end business applications based on SCA. The first part of this series (this article) is going to introduce both SCA and WID to the reader and illustrate the SCA building blocks in WID that may be used to create service-based enterprise applications. The second part is going to take an example of an enterprise application and explore the basic WID features that may be leveraged to create an application based on SCA.

Service Component Architecture (SCA)

As mentioned in the previous section, SCA describes a programming model for building and integrating applications that have been built using SOA.

Traditionally, the applications that have been created have a lot of dependency on middleware products and their associated integration with the application business logic. Application developers not only needed to know the development language but also had to be very well conversant with the middleware API's that they had to integrate with. And middleware comes in the form of databases, enterprise information systems (EIS), messaging, and EJB access, only to name a few. Each

ABOUT THE AUTHOR

Tilak Mitra is a Certified Senior IT Architect at IBM. He specializes in mid- to large- range enterprise and application architectures based on J2EE, MQ, and other EAI technologies. You can reach him at tmitra@us.ibm.com.

tmitra@us.ibm.com

middleware solves a particular problem space and has specific API's that needs to be well understood by the application developers. SCA aims at solving this dependency of middleware expertise by creating a service oriented interface around the various middleware technologies and protocols and provide a binding and wiring mechanism to allow developers to invoke the middleware layer functionality through a uniform service interface. This fosters loose coupling between application business logic and middleware technologies and allows for an inherent technology flexibility (of potentially changing the middleware or enhancing its capabilities) without changing the application code. This separation also allows the underlying technologies such as messaging and web services protocols to evolve with minimal impact on business services. SCA is also a model that promotes protection of business logic assets. It promotes the use of existing assets by providing bindings to a wide variety of technologies. SCA is a programming model that is designed to build SOA-based systems and is not a retrofit of technologies and development IDE's to support SOA. SCA provides dynamic composition capabilities by assembling services to provide higher level business capabilities and it does so in a way that can be adapted and changed as and when the business requirements changes or grows.

SCA is built around the premise that any SCA component provides a service interface that can be used by other SCA components. The interface invocation is done through wiring of components through their interface definitions. The service interfaces in SCA are defined using Web Service Description Language (WSDL) and Java interfaces. The implementation of the SCA components can be done by a variety of technologies, e.g. J2EE, CORBA, C++, etc. The binding protocols for service interface invocation can be WSDL, JMS, and Java Connector Architecture (JCA).

One more reason for using SCA is the adoption and subsequent adherence to the SOA starting from architecture, design right down to the development and integration of components, systems and business capabilities. SCA promotes the use of SOA in a manner that SOA-compliant artifacts are created and developed.

SCA changes the programming model from the use of disparate technologies at various layers (of Data, Invocation and Composition) to providing a single unified abstraction model at each layer of the programming stack. Whereas the traditional programming model uses various different technologies at each layer of a programming model, SCA provides a single technology at each layer of the stack. This allows for a much more consistent and simpler approach to programming as developers needs to learn only one technology for each layer in the programming stack. Figures 1 and 2 show the difference between the traditional programming model and the new programming model introduced by SCA.

SCA Artifacts

SCA is build around the notion of components that offer service oriented interfaces for external consumption and consume other services through service oriented interfaces.

SCA divides the building of SOA applications into two parts – *construction* of components that provide and consume services and *integration* of components to build business applications through wiring of services (a.k.a service composition). SCA provides a mechanism to package the components into modules. The packaging is based on loose coupling (to create different sca modules) and tight cohesion (to create components within a sca module).

The following are the high level SCA constructs:

1. ***Implementation*** **–** This is the business logic that can be written in any programming language e.g. Java, C++, BPEL, etc. When implementing the service, the focus is on the business purpose and less on the infrastructure technology. The implementation is a template. The same implementation can be reused multiple times and each instance of the implementation is called a *component*. The implementation can provide a *service*. A *service* is a set of operations encapsulated in an interface. When an implementation requires the usage of other services, they do so through the use of *references*. A reference is represented as an interface that abstracts the external service operations that the implementation is dependent upon. A *reference* is configured by binding the reference to the target service. An implementation may also have one or more *properties*. A property is a data value that can be configured external to the implementation. The property values can determine the business function of the implementation.
2. ***Component*** **–** This can be understood as an instance of an implementation together with the externally configurable *references* and *properties*. A component offers services and has dependencies on other components through *references*.
3. ***Service (SCA) Module*** **–** A service module is a collection of inter-dependent, high cohesive components that collectively offer higher-level services. The services offered by a service module are implemented by assembling (or wiring) together the components through their services and references. An SCA module contains components that are deployed together into a SCA runtime. The services offered by a SCA module are represented

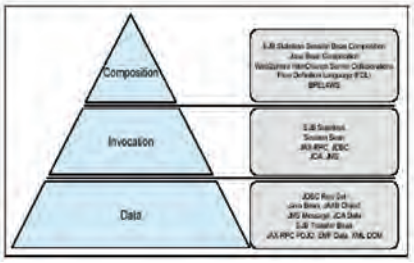


Figure 1: Traditional programming model

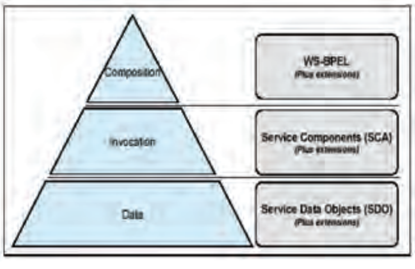


Figure 2: SCA programming model

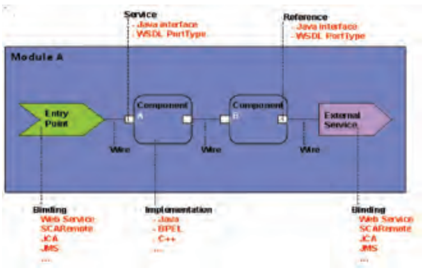


Figure 3: A simple SCA module

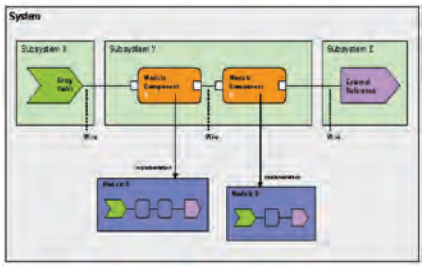


Figure 4: An SCA module in WID

as *EntryPoints*. The dependency of components (within a module) on services provided by other SCA modules is represented as *ExternalServices*. Similar to the way component *references* and *services* are represented through interfaces, the *EntryPoints* and *ExternalServices* are also represented through interfaces.

4. SCA Wires – Mention about bindings and the three types.

The SCA artifacts as described above need to be assembled to provide business functions.

SCA Assembly

There are two types of programming models with SCA: “micro service assembly” and “macro service assembly.” While the former is for the wiring of high cohesive components to create services, the latter is geared towards wiring of loosely coupled services in a bigger composition to realize higher level business processes through service assembly. While “micro service assembly” is achieved through module assembly, “macro service assembly” is achieved through system assembly.

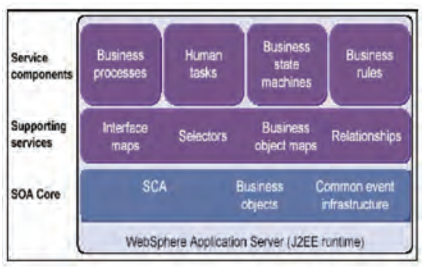


Figure 5: Architecture model for WebSphere Process Server

MODULE ASSEMBLY

The assembling i.e., wiring of components, reference and service interfaces within a module is known as module assembly. A component is wired through the references (to other component's implementation). The SCA wires are used to tie the references to the target implementations. The target implementation is exposed as a service. In this way, components may be wired in tandem to create a service module. Often times a component is dependent on an external service. In these situations, the components reference is wired to an external service through one of the supported bindings (Web Service, JCA, JMS, etc). The service offered by the module is exposed through EntryPoints (as discussed earlier). For a module that has multiple components, some of the components have services and references that are only used for consumption inside the service module. These are called “local” services as the boundary of the service is defined by the encapsulating service module. The figure below shows an example of a module assembly.

SYSTEM ASSEMBLY

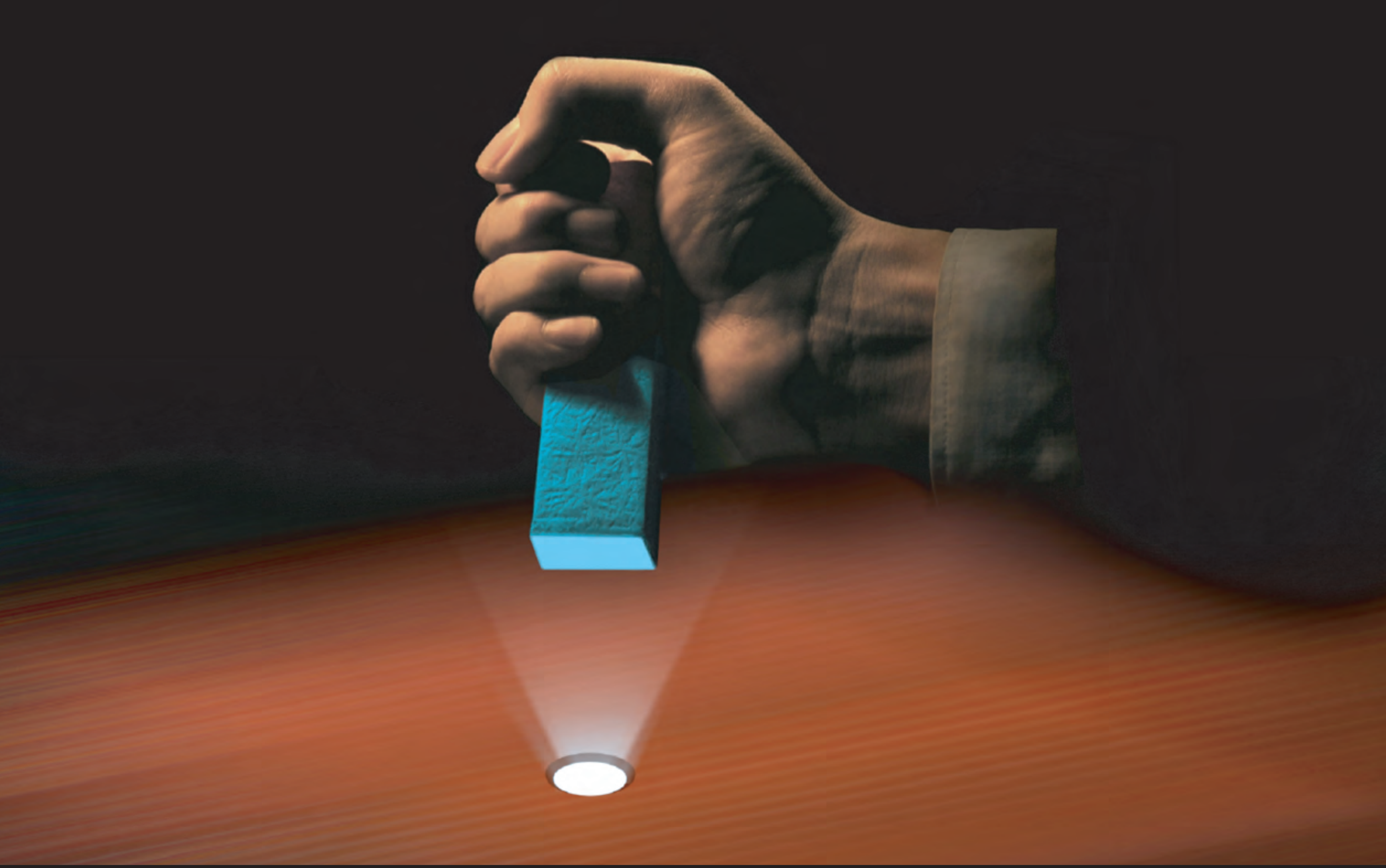
The wiring and composition of loosely coupled services to realize a high level business process is known as system assembly. A module EntryPoint may be used by other modules to consume the services of the offering module. EntryPoints may also be offered by an enterprise as services to consumers outside the enterprise boundary. When systems are developed from existing service modules, often times it is seen that a set of modules are dependent on each other. Such modules may be grouped into subsystems to facilitate packaging and deployment onto the SCA runtime.

It is essential to note that the mechanism for module assembly and for system assembly is essentially based on the same principle of wiring. While in module assembly, components are wired to one another through services and references, in system assembly the modules are wired together using entrypoints and externalservices. This lends itself to a consistent development and integration methodology for creation and composition of services in an SOA.

WebSphere Integration Developer

WebSphere Integration Developer (WID) V6.0 is an Eclipse-based Integrated

> “SCA is built around the premise that any SCA component provides a service interface that can be used by other SCA components”



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.

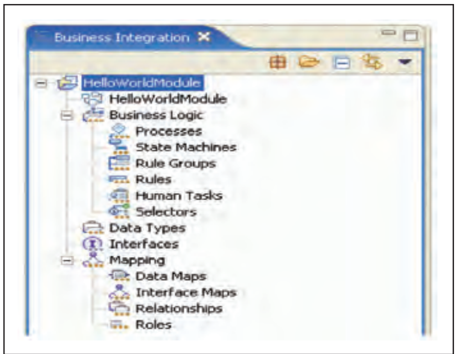
JReport
JINFONET SOFTWARE

© 2005. Jinfonet Software. All rights reserved. Jinfonet Software and JReport are trademarks or registered trademarks of Jinfonet Software, Inc. Crystal Reports is a trademark or registered trademark of Business Objects SA. All other marks mentioned herein may be the trademarks of their respective owners

“Services are invoked through operations that are exposed through the service interfaces”

Development Environment (IDE) that is used for developing and assembling applications based on SCA, which can be deployed on the IBM WebSphere Process Server V6.0 as the runtime.

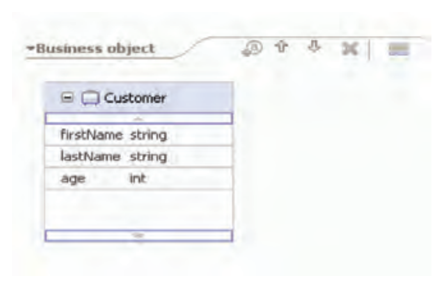
IBM WebSphere Process Server is the runtime and IBM WebSphere Integration Developer the design, development and integration tool that are amongst the latest of IBM product offerings to create SOA based solutions based on the SCA programming model.



Business Integration

- HelloWorldModule
  - HelloWorldModule
  - Business Logic
    - Processes
    - State Machines
    - Rule Groups
    - Rules
    - Human Tasks
    - Selectors
  - Data Types
  - Interfaces
  - Mapping
    - Data Maps
    - Interface Maps
    - Relationships
    - Roles

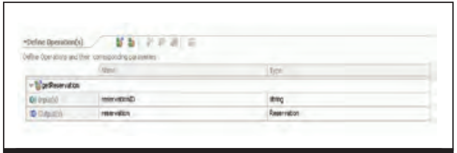
Figure 6: A sample system assembly (made up of multiple SCA modules)



Business object

Customer	
firstName	string
lastName	string
age	int

Figure 7: A Customer business object created in the Business Object editor



Define Operation(s)

Define Operations and their corresponding parameters

	Name	Type
getReservation		
Input(s)	reservationID	string
Output(s)	reservation	Reservation

Figure 8: An interface with operations (with inputs and outputs) created in the Interface editor

A good grasp of the SCA concepts and constructs are essential prerequisites for using WID for design and integration of SCA-based solutions. For those familiar with the existing Rational development tools of Rational Application Developer (RAD) and Rational Software Architect (RSA), WID is designed to run within RAD and RSA. WID installed to run within RAD and RSA essentially adds WID-specific workspace perspectives to the RAD and RSA. WID can also be installed as a standalone product.

The architecture model for WebSphere Process Server is depicted in Figure 5.

The figure depicts that WebSphere Process Server (WPS) is built on top of WebSphere Application Server. WPS provides service components like Business Processes, Human Tasks, Business State Machines and Business Rules that can be used in an overall integration solution. WID provides a development environment to create a SCA-based integration solution that may use any or all of the building blocks as depicted in the WPS architecture model.

The main high level construct in WID is a SCA module. Inside a module, all the architecture building blocks can be created for use within the module. Sometimes it is necessary to create some components the services of which need to be shared across multiple SCA modules. WID supports the creation of a *Shared Library* that capabilities of which can be used by multiple SCA modules. A sample SCA module is shown in figure 4.

The above figure illustrates the various SCA constructs that can be created in WID. Each module consists of a set of Business Logic artifacts, Data Types, Interfaces and Mapping elements. A brief overview of each of the constructs is given below:

- ***Processes* -** WS-BPEL compliant business processes can be created in WID. Business processes that are modeled in a modeling tool e.g. WebSphere Business Integration Modeler can also be imported into WID. WID supports both short and long running processes and it allows the implementation of business logic using BPEL.
- ***State Machines* -** is another way to model a business process. It is quite common to have business processes e.g. order processing, in which the state of the entity may vary based on conditions. A given state may be repeated during the process. These kinds of processes based on states and events are better modeled as a state machine than modeling them as processes (in general).
- Rule Groups and Rules - are a means for externalizing the business policies from the business logic implementation. This allows for dynamic changing of business rules by business analysts thereby providing better latency of changing business requirements. A rule group is a collection of similar rules.
- ***Human Tasks*** – are stand-alone components that can be used to assign work to

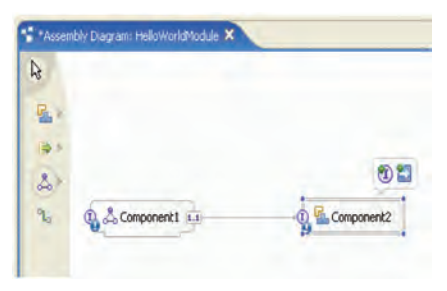


Figure 9: Snapshot of components created and wired in the Assembly Editor

Visit the *New*

www.SYS-CON.com

Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS

Stay ahead of the i-Technology curve with E-mail updates on what's happening in your industry

SYS-CON.TV

Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!

Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST

Streaming video on today's i-Technology news, events, and webinars

EDUCATION

The world's leading online i-Technology university

RESEARCH

i-Technology data "and" analysis for business decision-makers

MAGAZINES

View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES

Get all the news and information happening in other countries worldwide

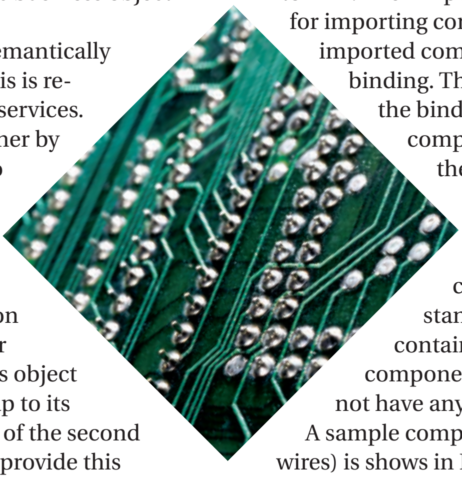
JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide
Information Storage+Security Journal
JDJ
Web Services Journal
.NET Developer's Journal
LinuxWorld Magazine
Linux Business News
Eclipse Developer's Journal
MX Developer's Journal
ColdFusion Developer's Journal
XML Journal
Wireless Business & Technology
Symbian Developer's Journal
WebSphere Journal
WLDJ
PowerBuilder Developer's Journal

SYS-CON MEDIA *The World's Leading i-Technology Publisher*

human resources (e.g. employees) or to invoke any other service. The Human Task service supports hierarchical escalation as well as a notification mechanism to inform about the escalation.

- ***Selectors* –** are a mechanism to select one of multiple service implementations that share the same service interface, based on a configurable policy/rule. This allows for two different implementations of the same service implementations to be invoked based on for example, a specific date or time.
- ***Relationships* –** is a mechanism to synchronize disparate representation of the same data in an enterprise. Often times, the same business entity e.g. Customer is represented in various different formats in different enterprise back end systems e.g. CRM, ERP, etc. Relationships provide a generic representation of the business entity identifier and then provide specific mapping of the same identifier to specific representations in back end systems. This provides a mapping technique between generic business object representations and application specific business object representations.
- ***Interface Maps* –** This allows the bridging of semantically similar by syntactically different interfaces. This is required when integrating existing systems and services. Interface maps translate one interface to another by mapping the parameters from one interface to the other.
- ***Business Object Maps* –** Business objects are representations of business entities. A business entity may have application-specific representations while a generic application neutral representation can also be defined. For application-application integration, a business object representation in one application needs to map to its corresponding business object representation of the second application. Business object maps are used to provide this mapping.



Services are invoked through operations that are exposed through the service interfaces. The parameters that can be passed into and out of the operations can be either simple/primitive Java data types (e.g. String, Integer, etc) or they can be complex types. The complex object types are called *Business Object*s (BO) in the SCA lingo. Business objects are an extension to the concept of *Service Data Objects* (SDO) specification.

WID Editors

WID provides three main editors: Business Object editor, Interface editor and Assembly editor.

Business Object Editor

The Business Object editor provides a drag and drop user interface to define new Business Objects, assign attributes and their data types. The Business Objects that are defined all reside under the "Data Types" folder for a given SCA module. An example of a Business Object created in the Business Object editor is shown in Figure 5.

Interface Editor

The Interface editor is used to define interfaces that will be subsequently used by components. The editor provides a drag and drop user interface to create all the elements on an interface i.e. operations and the inputs and outputs of each operation. The interfaces that are defined all reside under the "Interfaces" folder for a given SCA module. An example of an Interface created in the Interface editor is shown in Figure 6.

Assembly editor

The assembly editor is used to develop components and assign interfaces to implementation details to them. *Wiring* is the process of assembling components to create modules and system assemblies. Wiring just depicts the component dependencies in an assembly. Components defined in WID needs to be provided with an implementation. Components can also be *import*ed into WID. The "Import" feature in the assembly editor allows for importing components from outside the module. The imported components need to be assigned a specific binding. The imported components are invoked using the bindings that are generated for each imported component. Imported components can be used in the composition of a module assembly.

Applications that are not defined as SCA components (for example, JavaServer Pages, or JSPs) can still invoke SCA components; they do so through the use of stand-alone references. Stand-alone references contain partner references that identify the components to call. Alone, stand-alone references do not have any implementation or interface.

A sample composition (using components, interfaces and wires) is shows in Figure 7.

Figure 7 each component is first assigned one or more interfaces before its implementation can be defined. The components are then wired together in a module composition.

For the ease of testing, WID provides a lightweight version of the WPS runtime in which the applications may be deployed and tested before the final deployment onto a more robust WPS runtime. This facilitates unit testing of SCA-based systems, by providing a runtime environment right within the developer's environment. WID also provides a Universal Test Client (UTC) that may be used by developers to test individual components before testing the entire module.

Summary

This article gave an overview of Service Component Architecture (SCA) and how it facilitates true SOA-based application development. It also introduced us to the development environment, WID, and how it can be used for the design and development of SCA-based applications.

The next article in this series is going to take a sample SCA application and walk the reader through the actual creation of the application using the various features of WID.

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

CALL TODAY! 888-303-5282

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

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

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

□ **LinuxWorld Magazine**

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

□ **JDJ**

U.S. - Two Years (24) Cover: $144	You Pay: $99.99 /	Save: $45 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $89.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Web Services Journal**

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

□ **.NET Developer's Journal**

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

□ **Information Storage + Security Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $39
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ **Wireless Business & Technology**

U.S. - Two Years (12) Cover: $120	You Pay: $49.00 /	Save: $71 + FREE $198 CD
U.S. - One Year (6) Cover: $60	You Pay: $29.99 /	Save: $30
Can/Mex - Two Years (12) $120	You Pay: $69.99 /	Save: $51 + FREE $198 CD
Can/Mex - One Year (6) $60	You Pay: $49.99 /	Save: $10
Int'l - Two Years (12) $120	You Pay: $99.99 /	Save: $20 + FREE $198 CD
Int'l - One Year (6) $72	You Pay: $69.99 /	Save: $2

□ **MX Developer's Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ **ColdFusion Developer's Journal**

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

□ **WebSphere Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ **PowerBuilder Developer's Journal**

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

□ **WLDJ**

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
Can/Mex - Two Year (12) $120	You Pay: $69.99 /	Save: $50
Int'l - Four Years (24) $240	You Pay: $120 /	Save: $120 + FREE $198 CD
Int'l - Two Year (12) $120	You Pay: $79.99 /	Save: $40

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

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

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $^{\$}210^{00}$ Pay only $99 for a 1 year subscription plus a **FREE CD**

- 2 Year – $179.00
- Canada/Mexico – $189.00
- International – $199.00

6-Pack

Pick any 6 of our magazines and save up to $^{\$}340^{00}$ Pay only $199 for a 1 year subscription plus **2 FREE CDs**

- 2 Year – $379.00
- Canada/Mexico – $399.00
- International – $449.00

9-Pack

Pick 9 of our magazines and save up to $^{\$}270^{00}$ Pay only $399 for a 1 year subscription plus **3 FREE CDs**

- 2 Year – $699.00
- Canada/Mexico – $749.00
- International – $849.00

SYS-CON MEDIA

Two-Part Article

Building SOA Solutions with Service Component Architecture

BY ROLAND BARCIA AND JEFF BRENT

In the previous article (part 1 in *WebSphere Journal* vol: 3. iss: 4) we began to build an SCA project in WebSphere Process Server. Here in part 2 we pick up the discussion. To see the associated images, please view the article online at http://www.ibm.com/developerWorks/websphere.

d. The business object is defined using standard XML schema type. You can open the business object using an XML editor to see it. (You can open the business object in a text editor by right clicking it and selecting Open with => and the editor of your choice.)

e. Save and close the Business Object Editor.

f. The XML schema should look something like the code in Listing 1.

3. We will now create a second business object to represent the response.
   a. Create another business object following the same steps above. This business object will have three fields:
   - customerId of type string
   - creditScore of type int
   - creditLimit of type double

b. As mentioned earlier, you can change the type by selecting the type column as shown in Figure 19.

c. Save and close the Business Object Editor

You now have two business objects created.

Define the service interface

You are now ready to create your SCA Interface. Again, the credit approval service is a request-response service that receives a credit application and returns a credit rating synchronously. The service interface defines the interaction between the service client and the service provider.

There are several ways to create an interface. If you choose a Java interface, you can use the Java Eclipse tools within WebSphere Integration Developer to do this. In our example, we will create a WSDL interface from the Business Integration perspective. You can do this using the Assembly Editor or you can do it using the Business Integraiton view. We will use the latter. (We will use the Assembly Editor later to create the implementation.)

1. First, we will create the Interface using the Business Integration view menu.
   a. Right-click on the Interfaces menu item and select New => Interfaces as shown in Figure 20 (See online version).
   b. On the New Interface menu, enter the name CreditApproval. (Keep in mind that we are using the default package and folders in our example for illustration purposes. You can select a folder to easily group different interfaces by functionality.)

2. The CreditApproval Interface is a simple WSDL file. WebSphere Integration Developer comes with a simple WSDL editor that you can use to build up your interface.
   a. The Interface Editor should have opened when you created the Interface. If it is not open already, you can double click the interface in the business integration view to open it.
   b. We will first create a simple request-response operation. (You can also create one-way operations that can be used with asynchronous invocations, but for now, we are only creating a simple synchronous request.) Press the Add Request Response Operation icon as shown in Figure 22.

(This article was first published on developerWorks WebSphere at http://www.ibm.com/developerWorks/websphere.)

ABOUT THE AUTHORS

Roland Barcia is a Consulting IT Specialist for IBM Software Services for WebSphere in the New York/New Jersey Metro area. He is a co-author of IBM WebSphere: Deployment and Advanced Configuration..

Jeff Brent is an Advisory Software Engineer for the WebSphere Business Integration Competency Center (WBICC). His responsibilities include helping independent software vendors develop integration strategies for WebSphere products.

c. An operation will be created on the editor. Name the operation calculateCreditRating. (Figure 23)

d. Now we need to define the parameters. Remember, we created two business objects, one for input and the other for output. With the operation created, select the Add Input icon as shown in Figure 24.

e. Name the input CreditApplication.

f. Select the Type column and find the CreditApplicaiton business object. As you can see, the business object is now a valid type that you can use to build up the interface, as shown in Figure 25. (Optionally, you can select to create a new business object here.)

g. Next, select the Add Output icon as shown in Figure 26.

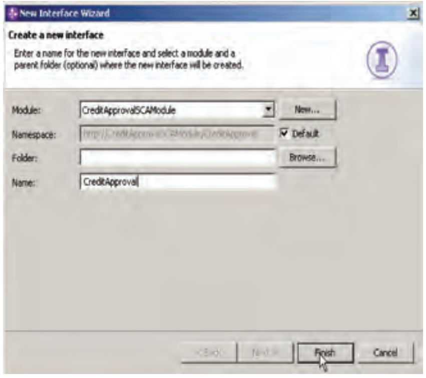
h. Select CreditRating as the type, similar to what we did with the input.

i. Save and close the Interface Editor.

j. If you wish to inspect the WSDL file, you can right click the CreditApproval.wsdl file in the Physical Resources view and open it with a text editor. (Figure 28)

The WSDL file should look like the code in Listing 2.

Generate the component and provide an implementationWe are now ready to create our SCA implementation. At this point, we have created standard interfaces and business objects. We will now define our SCA component. You will use the SCA Assembly Editor to do this.



New Interface Wizard

Create a new interface

Enter a name for the new interface and select a module and a parent folder (optional) where the new interface will be created.

Module: CreditApprovalSCAModule

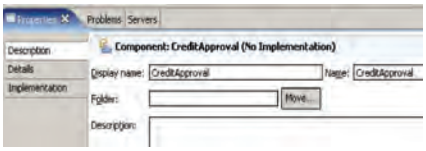
Namespace: http://CreditApprovalSCAModule/CreditApproval

☑ Default

Folder:

Name: CreditApproval

Figure 21: Create a new interface



Component: CreditApproval (No Implementation)

Display name: CreditApproval

Name: CreditApproval

Folder:

Description:

Figure 33: Change component name

1. First, we will define the SCA component.

   a. Open the SCA Assembly Editor by double clicking the CreditApprovalSCAModule menu item, as shown in Figure 29.

   b. The SCA Assembly Editor has a palette you can use to create SCA artifacts. You can also drag certain artifacts from various views. Simply drag the CreditApproval interface onto the Assembly Editor as shown in Figure 30. (Alternatively, you could also drag a Java component from the palette and associate the interface afterwards.)

   c. A text box will appear. Select Component with No Implementation Type as shown in Figure 31.

   d. You should now see an SCA component on the Assembly Editor named Component1. (Figure 32)

   e. You can change the name by selecting the component and typing the name, or you can change it using the Properties Editor as shown in Figure 33. Change the Display Name to CreditApproval. The Name field should change automatically.

2. We now have an SCA component with an interface, but no implementation. We will now use the Assembly Editor to generate the implementation.

   a. Generate a skeleton implementation by right-clicking the component in the Assembly Editor and selecting Generate Implementation => Java. (Figure 34)

   b. The new Java implementation should open in a Java Editor; you will see the calculateCreditRating. A simple code snippet is provided in the download file, in C:\SCA_ArticleSeries\Part1\CodeSnippet1.txt. The method is shown in Listing 3.

3. The code uses the SCA service manager to locate the Business Object Factory, which is used to create business objects from XML schemas. In our demonstration purposes, we create the Response Data Object and return hard coded data.

4. If you pasted in the code, you should have some compile errors. You can resolve the these by right clicking in the editor and selecting Source => Organize Imports as shown in Figure 35.

5. Save and close the Java file, but leave the Assembly Editor open.

Unit test the SCA componentWebSphere Integration Developer provides the ability to unit test components using a unit test tool. Depending on the type of SCA implementation, you can test SCA components in a J2SE environment, which enables you to test components without a full application server; of course, this greatly depends on the type of component you have. A Java implementation can be easily tested in a J2SE environment, but an SCA component realized by a BPEL flow would need a BPEL engine like WebSphere Process Server. In our example, we will use the Test Component feature to test our SCA component.

To launch the Test Component feature:

1. Right click the CreditApproval component in the SCA Assembly Editor and select Test Component. (Figure 36)
2. This will launch the Test Component Editor. On the right side of the tool, enter the test data as shown in Figure 37, then press Continue.
3. A list of available run times will display. Select Eclipse 1.4 JVM, then Finish. (Figure 38)
4. To begin the test, select the Return item as shown in Figure 39. Monitor the Events window to see the flow.
5. On the right side, you will see the result. (Figure 40)

Create SCA client Web application and testing with WebSphere Process Server1. To have an SCA component be invoked by a non-SCA component packaged in the same SCA deployment, we need to create a standalone reference:

a. In the SCA Assembly Editor, select the arrow next to the Import icon on the palette. This will expose a smaller submenu. Select the Standalone references icon and drag it anywhere

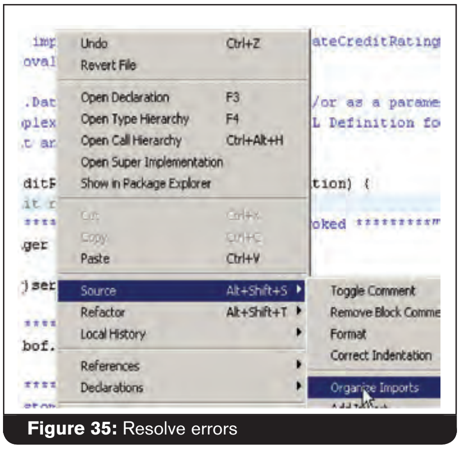


Figure 35: Resolve errors

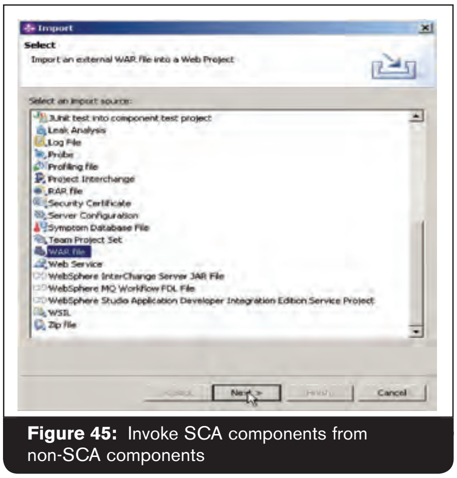


Figure 45: Invoke SCA components from non-SCA components

on the Assembly Editor as shown in Figure 41.

b. Click inside the Stand-alone References box (not the text) and drag it to the Credit Approval component as shown in Figure 42.

c. Press OK to create a matching reference. (Figure 43)

d. The next text box will give you an option to generate a Java interface. Remember, you created a WSDL Interface. However, WebSphere Integration Developer is generating an equivalent Java interface to let Java clients interact using the SCA API. Select Yes. (Figure 44)

2. With a standalone reference, we can now invoke SCA components from non-SCA components in the same module. Remember, an SCA module is realized by an EAR file, so you can add additional J2EE components to the SCA module. In our example, we are going to import an existing WAR file that has a JSP file. That WAR file will be part of the underlying EAR file.
   a. Select File => Import to bring up the Import wizard. Select WAR file and press Next. (Figure 45)
   b. Select C:\SCA_ArticleSeries\Part1\CreditApprovalClient.war. Unselect Add module to an EAR Project.
   c. Select Yes to switch to the Web perspective. (Figure 47)
   d. Open the JSP file to examine the code. You will find it under the Dynamic Web Projects directory, as shown in Figure 48. The JSP is called creditApprovalClient.jsp.
   e. We use a simple JSP to receive input and use the SCA API to invoke an SCA component. (Keep in mind this JSP does follow best practices for using MVC).

3. SCA has a Dependency Editor that enables you to add different packages to the SCA module:
   a. Open the Dependency Editor by right-clicking the CreditApprovalSCAModule and select Open Dependency Editor. (Figure 49)
   b. Expand the J2EE section, then press the Add button. (Figure 50)
   c. Select the CreditApprovalClient Web project as shown in Figure 51.
   d. Be sure to unselect On Build Path. (Build Path adds this component to the classpath of the SCA module. In our case, we need to make the WAR file depend on the SCA module.) Save the Dependency Editor. This makes the WAR file part of the EAR file.
   e. To make the Web project see the SCA Java interface, right click the Web project and select Properties.
   f. Go to Java JAR Dependencies and select CreditApprovalSCAModuleEJBClient.jar, which is the generated EJB client JAR file that is generated underneath. (Figure 53)
   g. Close the editor. At this point, you can choose to perform a full build. This may not be necessary, but we always like doing a full build before deploying. If you have automatic build, you can do a Clean as shown in Figure 54.

4. We are now ready to deploy and test our SCA component and Web client on WebSphere Process Server. Remember, our SCA module is realized by a J2EE application. You can switch to the J2EE perspective to view the artifacts. Keep in mind that these are generated artifacts. (We will cover these later in this article series.)
   a. Switch to the Servers view on the bottom of the Business Integration perspective. Right click WebSphere Process Server and select Start as shown in Figure 56.
   b. Wait for the server to start. Check the console and wait for the messages shown in Figure 57.
   c. Right-click the server again and select Add and remove projects. (Figure 58)
   d. Select CreditApprovalSCAModuleAdd and select Add as shown in Figure 59. This will move it into the Configured projects box on the right.
   e. Monitor the progress on the lower right corner and wait for the application to be deployed. (Figure 60)
   f. Verify at the console that the CreditApporvalSCAModuleApp is started. (Figure 61)
   g. Switch to the Web perspective. Right-click on the JSP and select Run => Run on Server as shown in Figure 62.
   h. Choose the existing WebSphere Process Server and select Set server as project default, then Finish. (Figure 63)
   i. When the application executes, enter the application test data shown in Figure 64.
   j. You should get the results shown in Figure 65.
   k. Close the browser and stop the server.

Congratulations, you have just created and tested your first SCA component!

Service Component Architecture (SCA) is a major shift in service-oriented architectures (SOA). SCA gives us a programming model for SOA, and will serve as a foundation for integration for years to come. Stay tuned for upcoming articles in this series, which will explore details of specific SCA features and the benefits they can bring to your applications.

www.linuxworld.com

Connect

Subscribe Today!

Connect online for fastest service... don't miss another issue of LWM!

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY $49.99 12 ISSUES/YR

LOG ON TO

SYS-CON MEDIA

The World's Leading i-Technology Publisher

Connect

www.linuxworld.com

WebSphere JOURNAL Advertiser Index...

COMPANY	URL	PHONE	PG
AJAX Seminar	http://www.ajaxseminar.com/	201-802-3022	37
Parasoft	www.parasoft.com/jdjmagazine	888-305-0041	52
Instantiations	www.instantiations.com/analytix	800-808-3737	2
ISSJ	http://issj.sys-con.com/	888-303-5282	43
IT Solutions guide	http://itsolutions.sys-con.com/	201-802-3021	25
iTVCON	www.iTVcon.com	201-802-3021	49
ITKO	www.itko.com	877-BUY-ITKO	51
Java developer's journal	http://jdj.sys-con.com/	201-802-3021	47
Jinfonet	www.jinfonet.com	301-838-5560	15
Motive	www.motive.com/within1	512-531-2527	7
Nastel	www.nastel.com/uncomplicate	800-580-2344	11
Prolifics	www.prolifics.com	800-675-5419	3
sys-con media website	www.sys-con.com	888-303-5282	17
Tangosol	www.tangosol.com	617-623-5782	9
webservices edge/open source	www2.sys-con.com/events	201-802-3066	29
websphere journal	www.sys-con.com	888-303-5282	41
Wily	www.wilytech.com	888-GET-WILY	5

REPRINT IT!

Once you're in it...

...reprint it!

- ColdFusion Developer's Journal
- Java Developer's Journal
- Web Services Developer's Journal
- Wireless Business & Technology
- XML Journal
- PowerBuilder Developer's Journal
- .NET Developer's Journal

Contact Dorothy Gil
201 802-3024
dorothy@sys-con.com

REprints

SYS-CON MEDIA

Listing 1:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
targetNamespace=”http://CreditApprovalSCAModule”>
  <xsd:complexType name=”CreditApplication”>
    <xsd:sequence>
      <xsd:element minOccurs=”0” name=”customerId” type=”xsd:string”/>
      <xsd:element minOccurs=”0” name=”firstName” type=”xsd:string”/>
      <xsd:element minOccurs=”0” name=”lastName” type=”xsd:string”/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Listing 2:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<wsdl:definitions xmlns:bons1=”http://CreditApprovalSCAModule”
xmlns:tns=”http://CreditApprovalSCAModule/CreditApproval”
xmlns:wsdl=”http://schemas.xmlsoap.org/wsdl/” xmlns:xsd=”http://www.
w3.org/2001/XMLSchema”
name=”CreditApproval” targetNamespace=”http://CreditApprovalSCAModule/
CreditApproval”>
  <wsdl:types>
    <xsd:schema targetNamespace=”http://CreditApprovalSCAModule/
CreditApproval”
xmlns:bons1=”http://CreditApprovalSCAModule”
xmlns:tns=”http://CreditApprovalSCAModule/CreditApproval”
xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
      <xsd:import namespace=”http://CreditApprovalSCAModule”
schemaLocation=”xsd-includes/http.CreditApprovalSCAModule.xsd”/>
      <xsd:element name=”calculateCreditRating”>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name=”CreditApplication” nillable=”true”
type=”bons1:CreditApplication”/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name=”calculateCreditRatingResponse”>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name=”CreditRating” nillable=”true”
type=”bons1:CreditRating”/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
    <wsdl:message name=”calculateCreditRatingRequestMsg”>
    <wsdl:part element=”tns:calculateCreditRating”
name=”calculateCreditRatingParameters”/>
  </wsdl:message>
    <wsdl:message name=”calculateCreditRatingResponseMsg”>
    <wsdl:part element=”tns:calculateCreditRatingResponse”
name=”calculateCreditRatingResult”/>
  </wsdl:message>
    <wsdl:portType name=”CreditApproval”>
    <wsdl:operation name=”calculateCreditRating”>
      <wsdl:input message=”tns:calculateCreditRatingRequestMsg”
name=”calculateCreditRatingRequest”/>
      <wsdl:output message=”tns:calculateCreditRatingResponseMsg”
name=”calculateCreditRatingResponse”/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

Listing 3:

```
public DataObject calculateCreditRating(DataObject creditApplication) {
//Create and return a credit rating object.
  System.out.println(“********* credit approval service invoked ******”);
  ServiceManager serviceManager = new ServiceManager();

  BOFactory bof = (BOFactory)serviceManager.locateService(“com/ibm/web-
sphere/bo/BOFactory”);
  System.out.println(“********* BOFactory created ******”);
  DataObject creditRating = bof.create(“http://CreditApproval”,
“CreditRating”);

  System.out.println(“********* credit rating object created ******”);
  creditRating.setString(“customerId”, creditApplication.getString(“custome
rId”));

  creditRating.setInt(“creditScore”, 750);
  creditRating.setDouble(“creditLimit”, 10000d);
  System.out.println(“********* returning credit rating object ******”);

  return creditRating;

}
```

Reach Over 100,000

Enterprise Development Managers & Decision Makers with...

>Linux >Java >Web Services >.NET >XML >Wireless >Storage >Security

The Premier Resource for Today's Corporate & IT Decision Makers

VOL. 1 ISSUE: 2

IT solutions >GUIDE

WWW.SYS-CON.COM/IT

Finding ROI

Profiting from Today's i-Technologies

Top IT Solution Providers!

- > CrossTec Corporation
- > Enerjy Software
- > EV1 Servers
- > FusionWare
- > Global Knowledge
- > Gluecode Software
- > Itemfield
- > JadeLiquid Software
- > Parasoft
- > PointBase
- > Promind Systems
- > Quadbase Systems
- > Wily Technology

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!

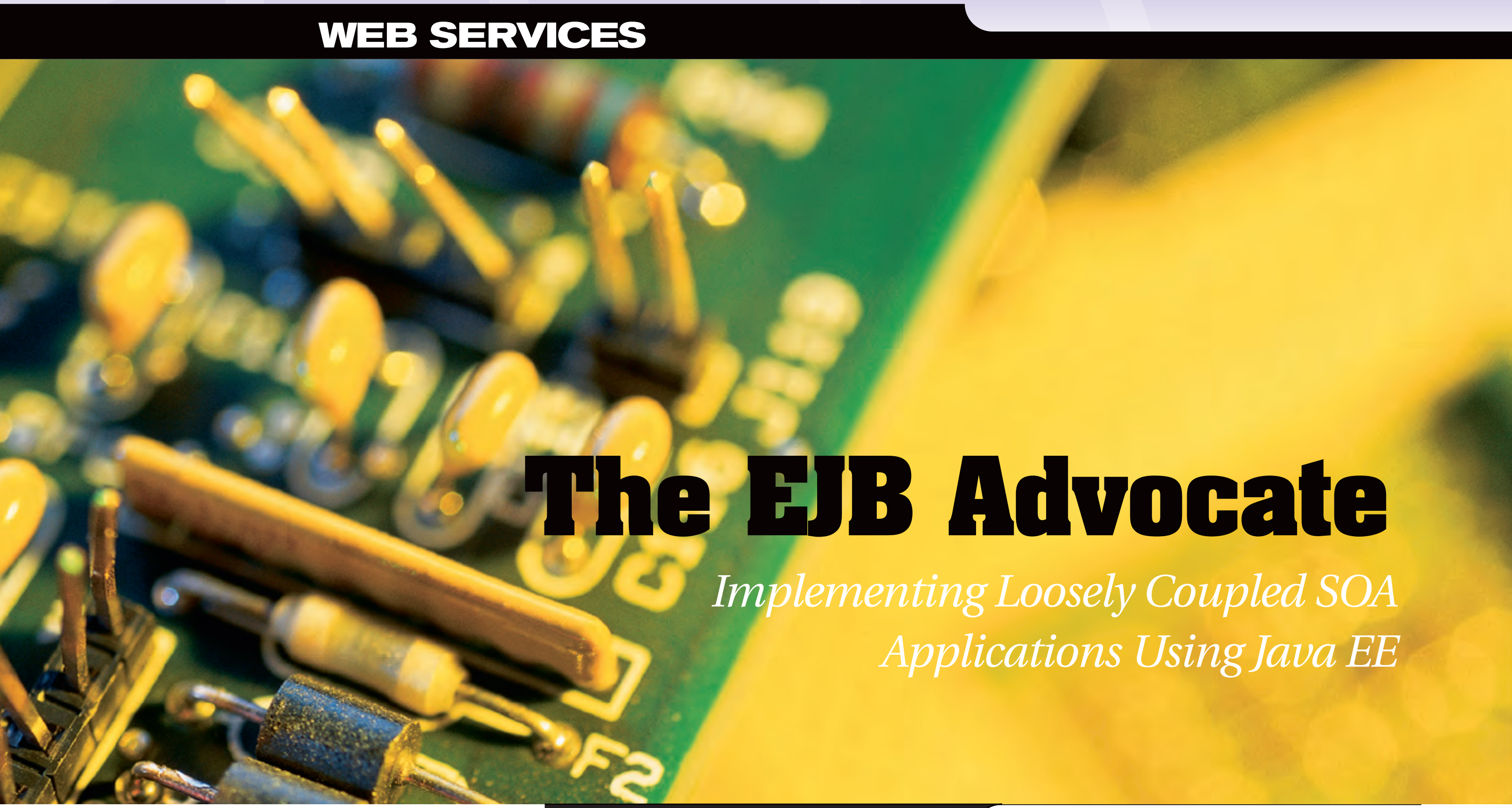
Get Listed as a Top 20* Solutions Provider

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 EJB Advocate

Implementing Loosely Coupled SOA Applications Using Java EE

BY GEOFF HAMBRICK

The EJB Advocate explores various forms of loose coupling in service-oriented architecture, with a focus on when it is best to take advantage of the asynchronous processing power provided by message-driven beans.

ABOUT THE AUTHOR

Geoff Hambrick is a lead consultant with the IBM Software Services for WebSphere Enablement Team and lives in Round Rock, Texas (near to Austin). The Enablement Team generally helps support the pre-sales process through deep technical briefings and short term Proof of Concept engagements. Geoff was appointed an IBM Distinguished Engineer in March of 2004 for his work in creating and disseminating best practices for developing J2EE applications hosted on IBM WebSphere Application Server.

In each column, The EJB Advocate presents the gist of a typical back-and-forth dialogue exchange with actual customers and developers in the course of recommending a solution to an interesting design issue. Any identifying details have been obscured, and no "innovative" or proprietary architectures are presented. For more information, see Introducing the EJB Advocate at the IBM developerWorks website.

Is your definition of loose coupling too tight?

Since this is the last article of 2005, this exchange about Java Platform, Enterprise Edition (Java EE) components other than session and entity beans seemed like a good way to summarize the year-long discussion conducted through this column and put the all of the components together into a complete service oriented architecture.

(The following article was originally published in The IBM WebSphere Technical Journal at the IBM developerWorks website. The original can be located at www.ibm.com/developerworks/)

The problem: too much focus on sessions and entities in SOA

Dear EJB Advocate,

So far your columns have been all about session and entity EJBs for services, which is great for directly connected Java applications, like HttpServlets for Web based clients, or Swing applications for rich (we don't like to say "fat") clients. But we have heard that service-oriented architecture is all about loose coupling. Doesn't that imply using SOAP to provide language neutrality and asynchronous protocols that enable the client and server applications to run independently as much as possible? In other words, why haven't you talked much about JMS and message-driven beans?

Signed,
Feeling Disconnected

There are many aspects to loose coupling to consider

Dear Disconnected,

The EJB Advocate has focused on the

services layers of the application and not much at all on the client side because I am a big believer in the old adage *form follows function*.

The reason is that many projects embarking on SOA fail because they get caught up in the implementation details before they have a good model for defining services in the first place.

This tendency is somewhat natural because most of the people I deal with on the subject of SOA are architects and programmers who know that the devil is always in the details and want to get down to them as quickly as possible. So, as soon as we agree that the attributes of a good service are that they are coarse grained, stateless, mediatable and adaptable (see Is it ever best to use EJB components without facades in service oriented architectures?), it seems obvious that session beans with data transfer objects will play a role in the implementation.

But that previous column leads one to question whether a session bean is necessary at all, introducing the possibility of using entity beans and their Home methods instead of session beans. Figure 1 shows both approaches being used simultaneously.

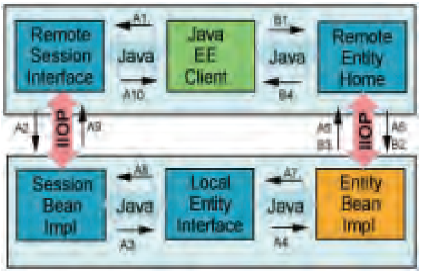


Figure 1: A service implemented with session and entity EJBs waiting to be used

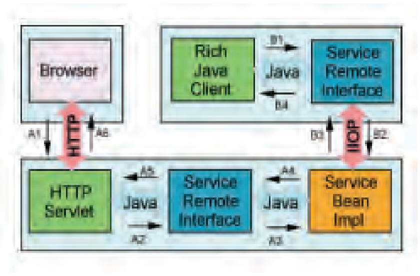


Figure 2: Service deployed locally to a Web application and remotely to a rich

Figure 1 shows how the pure entity approach, enabled since EJB 2, has fewer components and a shorter path length when using session beans that pass through. A green box is used to represent the client, blue ones to represent various interface and facade classes. The orange box is the commonly accessed entity bean. The boxes are connected with double headed arrows labeled with the protocol used to communicate between the components; the blue ones representing Java calls in the same JVM, and the red ones representing connections that are remote (using RMI/IIOP in this case). The flow arrows are numbered to show the end-to-end flow, with A1-A10 showing the flow from the Java client through the session bean to the entity and back, and B1-B4 showing the flow from the client directly to the entity bean and back.

The programming model to retrieve the service interface used by the client is simpler too, although this is not shown in the diagram. Retrieving a session bean interface requires looking up the session Home in a JNDI context and using it to create a session; an entity Home merely needs a lookup, its methods can be directly invoked without creating a reference to an EJB Object. The following two code samples show the difference.

Figure 1 and the associated code samples show a real benefit of the Java EE programming model, whether or not you choose to use the entity Home method approach. The programming model changes incrementally and provides *backwards compatability*. In short, your best practices can evolve without forcing you to change existing applications. Also, the JNDI context provides a related aspect of loose coupling that should not be overlooked: *implementation independence*.

Remote interfaces to session or entity EJB components provide *location independence*. Using a remote interface makes it possible to deploy the client application and service components in the same JVM where it makes sense for the response time, throughput, and maintainability goals of the system, such as with a Web application using HttpServlets. Figure 2 shows just such a configuration, where an enterprise-quality application server, like IBM WebSphere Application Server, "short circuits" a remote interface to use Java and pass by reference when the client and service components are co-deployed (whether the service is implemented as an entity Home method or session bean). Flows A1-A6 show the use of an HttpServlet co-deployed with a service component. Flows B1-B4 show how it is reused by a remote rich client Java EE application.

But it sounds like you have decided that the most important aspects of loose coupling are *language neutrality* and *asynchronous operations*. And you are right that needing asynchronous operations will lead you towards using message-driven beans (MDBs) and JMS on the server side.

The approach that many programmers use when implementing a MDB is to invoke the EJB component representing the service through its remote interface to maximize location independence as described above.

Then, whether the service implementation is deployed remotely or locally, the

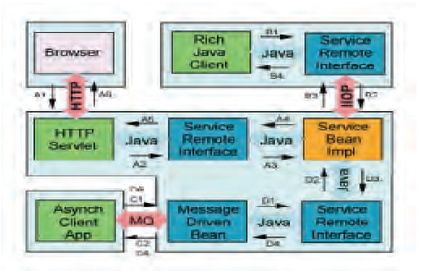


Figure 3: The service deployed with MDBs providing an asynchronous client channel

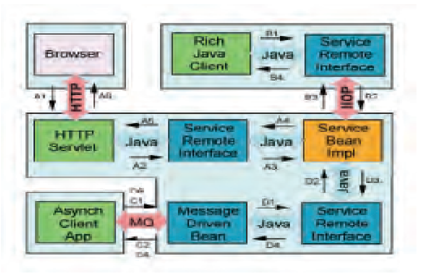


Figure 4: Separating the concerns of language neutrality from asynchronicity

purpose of the MDB is to serve as a simple adaptor, connecting an MQ layer into the EJB container where the service is hosted. JMS may be used by the asynchronous client application (if it is Java) or by the MDB (usually for the reply queue). Figure 3 shows this configuration as yet another input channel for our service implementation.

The flow C1-C2 is shown separately from D1-D6 to illustrate the independence of the client and server processes. C2 and D6 simply show an "acknowledgement" to the writer that the message was written and does not imply waiting. Listing 3 shows a typical structure of an MDB that should help clarify the processing that it must do:

The EJB Advocate would now like to get back to the issue of language neutrality that you brought up. It seems that you have tightly coupled the requirement of language neutrality with that for asynchronous processing. There is no reason that you cannot separate these concerns; the ability to parse a SOAP message and use it to invoke a session bean should be independent of whether the processing of that message is asynchronous (through MQ or another protocol over which JMS equivalent messages flow), or synchronous (such as through HTTP or even IIOP). In fact, some of the early "inventions" of Web services on Java EE applications used an HttpServlet to parse an XML message passed over HTTP. This approach eventually evolved to be SOAP/HTTP. Figure 4 shows yet another path that can be provided on top of the services implemented by an EJB component.

The Web Service Servlet and the Message Driven Bean could share the code that parses the data transfer objects from a stream extracted from the message string or HttpServletRequest. Similarly, the response or reply could share the code that generates a stream from the data transfer object (which could be an instance of Exception).

Hope this helps you understand the positioning of the Java EE components, all of which provide some form of loose coupling essential to a service-oriented architecture.

OK then,

Your EJB Advocate

Still too many options to choose from

Dear EJB Advocate,

Thanks.

I hadn't considered before now that services like JNDI and remote interfaces provide aspects of loose coupling. I can also see how we (as you put it) "tightly coupled" the idea of SOAP and MQ, and should try to separate them where possible. So it makes a lot of sense to treat the parsing and generating of SOAP messages like a service itself that is reused by the Web services servlet and the MDB.

But no thanks.

Before this discussion, SOA seemed pretty simple: every service exposed a SOAP/MQ interface. Now it seems that I have too many choices to think about, and since I am treating the SOAP message parsing and generation like services, why wouldn't I want to make a separate session bean to encapsulate them for reuse like you show in your figures?

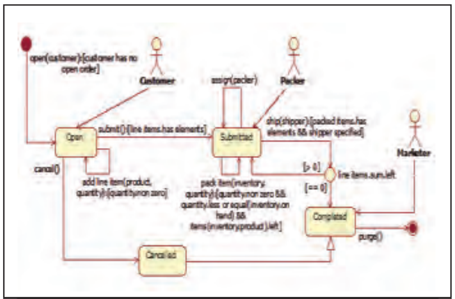

open(customer)[customer has no open order]
Customer
assign(packer)
Packer
Open
submit()[line items.has elements]
Submitted
ship(shipper)[packed items.has elements && shipper specified]
Marketer
cancel()
add line item(product, quantity)[quantity.non zero]
pack item(inventory, quantity)[quantity.non zero && quantity.less or equal(inventory.on hand) && items(inventory.product).left]
[> 0]
[== 0]
line items.sum.left
Completed
purge()
Cancelled

Figure 5: An example state-transition diagram showing the lifecycle of an order

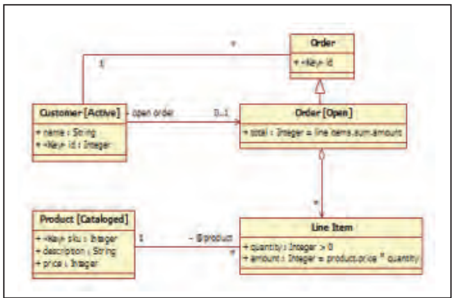


Figure 6: An example class diagram showing the "shape" of an open order

I'm afraid that I am:
Still Disconnected

Not everything is a service: use the business model to decide

Dear Disconnected,

This discussion is a great example of what happens when you get down to the implementation details too soon. The EJB Advocate column, "*Which type of EJB component should assemble the data returned by a service?*" was important because it describes a top-down approach to defining services that tied them directly to the business process model. To summarize the essential details of that approach here:

1. Develop a model of the business process, showing the major milestones in the lifecycle of an important domain object
   a. We used a state-transition diagrams for this model, where the states represent the milestones, and the transitions represent events that cause changes in that state. The transitions can be considered the services offered by the application (see Figure 5 for an example).
   b. We extend the state-transition diagram with UML "Actor" notation to show the owner of the object when it is in that state. The owner of the state is responsible for initiating the transitions, so drives the security model of the application (also see Figure 5).
   c. For each state in the business process, we also model the attributes of each domain object and the relationships between them needed to support the actions of the transitions. The notation we typically use is a UML class diagram (see Figure 6 for an example). Having separate diagrams for each state let us model the changing "shape" of the object over time. These diagrams drive the persistent data.

2. Develop a model of the user interface flow, showing how a given actor from the business process will interact with the system during a typical "session."
   a. Like the business process model, we used a state-transition diagram, where the states indicate the screens and dialogs and the transitions represents the actual UI events, like menu items being

ENGAGE AND EXPLORE...

The Technologies, Solutions and Applications that are Driving Today's Initiatives and Strategies...

CALL FOR PAPERS NOW OPEN!

SOA WebServices Edge 10th International conference+expo Edge 06

The Sixth Annual SOA Web Services Edge 2006 East - International Web Services Conference & Expo, to be held June 2006, announces that its Call for Papers is now open. Topics include all aspects of Web services and Service-Oriented Architecture

Suggested topics...

- > *Transitioning Successfully to SOA*
- > *Federated Web services*
- > *ebXML*
- > *Orchestration*
- > *Discovery*
- > *The Business Case for SOA*
- > *Interop & Standards*
- > *Web Services Management*
- > *Messaging Buses and SOA*
- > *Enterprise Service Buses*
- > *SOBAs (Service-Oriented Business Apps)*
- > *Delivering ROI with SOA*
- > *Java Web Services*
- > *XML Web Services*
- > *Security*
- > *Professional Open Source*
- > *Systems Integration*
- > *Sarbanes-Oxley*
- > *Grid Computing*
- > *Business Process Management*
- > *Web Services Choreography*

CALL FOR PAPERS NOW OPEN!

2006 ENTERPRISE> OPENSOURCE CONFERENCE+EXPO

The first annual Enterprise Open Source Conference & Expo announces that its Call for Papers is now open. Topics include all aspects of Open Source technology. The Enterprise Open Source Conference & Expo is a software development and management conference addressing the emerging technologies, tools and strategies surrounding the development of open source software. We invite you to submit a proposal to present in the following topics. Case studies, tools, best practices, development, security, deployment, performance, challenges, application management, strategies and integration.

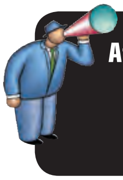
Suggested topics...

- > *Open Source Licenses*
- > *Open Source & E-Mail*
- > *Databases*
- > *ROI Case Studies*
- > *Open Source ERP & CRM*
- > *Open-Source SIP*
- > *Testing*
- > *LAMP Technologies*
- > *Open Source on the Desktop*
- > *Open Source & Sarbanes-Oxley*
- > *IP Management*

Submit Your Topic Today! events.sys-con.com

Sponsored by

WebServices Journal · XML Journal · .NET Developer's Journal · eclipse) developer's journal · WebSphere Journal · information Storage+Security journal · wldj · JDJ · LinuxWorld Magazine · MX developer's journal · asp.netPRO · SD Times · CoDe · Software Test & Performance



Attention Exhibitors:

An Exhibit-Forum will display leading Web services and OpenSource products, services, and solutions

*Call for Papers email: events@sys-con.com

For Exhibit and Sponsorship Information ► **Call 201 802-3023**

Produced by SYS-CON EVENTS

© 2005 WEB SERVICES EDGE. ALL RIGHTS RESERVED

selected and buttons being pressed. The actions associated with the transitions are specified in terms of business process transitions.

b. Also like the business process model, we build a class diagram to show the data that must be visible in each state to support the various choices. The implication is that this data must be derivable from user inputs and drives the read operations on the services in a top down fashion.

c. Unlike the business process model, we need not extend the state diagram with Actor notation, since the diagram itself can be thought of as a "day in the life" of a single user role.

This comprehensive approach insures that:

- The right operations are grouped together into a service (all the transitions and read methods associated with a state in the business process lifecycle).
- The purpose of each service is well understood (the actions are specified in terms of the related business objects).
- The pre-conditions needed to invoke a service and the post conditions that can result are communicated (the current state and optional guards, as well as the next state are specified by the transition).
- The user role responsible for invoking the service is identified (an actor is associated with each state).

None of this information is provided by a method signature regardless of where they are implemented, but this information is crucial to a good SOA. Otherwise, programmers will fall back on another tendency: *when in doubt, build it over again*. Since development costs are higher for SOA (to provide all the different mediators and adaptors needed for the completely loose coupling shown in Figure 4), this tendency against reuse can minimize the benefits.

To answer the issue in your reply about simplicity, none of this information forces you to expose the interfaces in a certain way, and the EJB Advocate has learned that simplicity is in the eye of the beholder. If you want to eliminate choices for the service developer, you can simply provide all of the “blue” components shown Figure 5 for each service:

- A remote service interface to provide location independent synchronous Java EE client access to the service operations.
- A MDB associated with each operation that provides asynchronous non-Java client access over a JMS compliant MQ implementation. Optionally, this or a different MDB can be coded to expect Java messages for JMS clients (to avoid the HTTP Parsing overhead).
- A Web services servlet associated with each operation that provides synchronous SOAP over HTTP client access. For those worried about the number of unused components that this approach would generate, another approach to simplicity is to apply what the EJB Advocate likes to call client-oriented architecture (COA); give the client exactly what they need to use the service in a way most natural to them.

This COA approach requires looking at the details of the business process and UI models to pick the most likely candidates for each approach. For example:

- Transitions between states in the business process lifecycle are likely candidates for asynchronous services, since there will be a change of "owner" for the associated domain object. For example, the submit method can simply change the state of the order to "submitted" in the open order application (we called this OrderEntry in the example above), and send a JMS message to copy it into the submitted order application (we called this one OrderFulfillment).
- Transitions within states should normally be synchronous, since there is no change of owner. As an example of why these operations should not be asynchronous, imagine if you went to a book seller's Web application and had to poll or wait for a pub-sub event for the catalog to be displayed or an item to be added to the shopping cart! And for those that want to use a pseudo-synchronous style over an asynchronous channel, please, please, please, see Bobby Woolf's book on Designing Messaging Systems.
- Only provide SOAP over HTTP or MQ for integration scenarios where you have non-Java clients or services involved.

The COA approach results in the components being developed “just-in-time,” which is why the EJB Advocate likes to recommend it.

And a final point to address your question (why not treat everything like a service, even the transformations associated with mediators and adapters): the simple answer goes beyond the fact that there can be too much of a good thing. When developing SOA or COA Java EE applications, it is best to consider services as operations on the business process model. Services have to do with the functional requirements of the application. Mediations and the associated transformations are associated with the non-functional requirements like reliability, usability, efficiency, maintainability, and portability. If you treat the mediations or the associated transformations as a first class service, you eventually obscure the real purpose of the application.

I know this is a lot to take in, so do not hesitate to contact me on the details as you apply these approaches.

OK then,
Your EJB Advocate

Conclusion

The dialog in these exchanges show how Java EE provides a complete implementation framework for applications employing service-oriented architecture, with each component or API playing an important role in some aspect of “loose coupling”:• **Operating system independence** is provided by Java itself because Java provides a write-once run anywhere language for components that decouples your code from the underlying operating system.

- Implementation independence is provided by the Java Naming and Directory Interface (JNDI), with the ability to bind a logical name to an implementation at runtime.
- Location independence is provided by remote interfaces using RMI over IIOP to stateless session beans or entity home methods

that encapsulate the services. RMI/IIOP is a stateful connection that is relatively fast, but does not scale all that well.

- Web server independence is provided by HttpServlets that can respond to the synchronous HTTP protocol. Unlike RMI/IIOP, HTTP is (normally) a stateless protocol that scales well, but does not perform as well due to the size of the messages and the need to create and break down the connection between client and server each time.
- Application independence is provided by the asynchronous Java Messaging Service (or MQ for non-Java clients) and message-driven beans (for the queue handler).
- Language independence is provided by using a standard message format like SOAP, whether it flows on RMI/IIOP, HTTP, or MQ. Thus, SOAP may be used by every component, but it is important to only use it when necessary.

We are willing to bet you can find others. The challenge is managing the tradeoffs associated with each. That should keep you busy until next year.

Resources

- Java Platform, Enterprise Edition (JEE) specification (formerly J2EE); the '2' was dropped in order to separate the main concept of a Java platform for the enterprise from a specific version.
- Enterprise Java Programming with IBM WebSphere, Second Edition by Kyle Brown, Gary Craig, Greg Hester, Russell Stinehour, W. David Pitt, Mark Weitzel, JimAmsden, Peter M. Jakab, Daniel Berg. Foreword by Martin Fowler.
- Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions by Gregor Hohe and Bobby Wolfe. Has a number of really great patterns and anti-patterns associated with messaging applications.

Listing 1.

```
Locating and invoking a remote session EJB meth-
odContext initCtx = new InitialContext();
Object obj = initCtx.lookup("java:comp/env/ejb/
OrderEntry");
OrderEntryHome home = (OrderEntryHome)PortableO
bjectRemote(
  obj, OrderEntryHome.class
);
OrderEntry ref = home.create();

// Method must be invoked on a session reference
CustomerData data = ref.getOpenOrderForCustome
r(cID);
```

Listing 2.

```
Locating and invoking the equivalent remote
entity EJB Home methodContext initCtx = new
InitialContext();
Object obj = initCtx.lookup("java:comp/env/
Customer");
CustomerHome ref = (CustomerHome)PortableObjec
tRemote(
  obj, CustomerHome.class
);

// Note how the method is invoked directly
CustomerData data = ref.getOpenOrder(cID);
```

Listing 3.

```
A typical message-driven bean implementation-
public class OrderSubmitMsgHandler implements
MessageDrivenBean {
    private transient MessageDrivenContext con-
text = null;
    // Interface supported by either the ses-
sion Bean or entity Home
    private transient OrderEntry ref;

    public void ejbCreate() {
        // Cache the home according to either
snippet one or two
    }
    public void ejbRemove() {}
    public void setMessageDrivenContext(MessageD
rivenContext mdc) {
        context = mdc;
    }
    // Message acknowledge and database update
participate
    public void onMessage(Message inMessage) {
        // Parse the customer from the message
  try {
      ref.submit(customer);
  }
        catch (CustomerDoesNotExist e) {
            // Send JMS message to reply queue
from exception
        }
        catch (OrderNotOpen e) {
            // Send JMS message to reply queue
from exception
        }
        catch (OrderHasNoItemsException e) {
            // Send JMS message to reply queue
from exception
        }
    }
}
```

THREE REASONS TO

blog-n-play.com

1 Get instantly published to 2 million+ readers per month!

blog-n-play™ is the only **FREE** custom blog address you can own that comes with instant access to the entire i-technology community. Have your blog read alongside the world's leading authorities, makers and shakers of the industry, including well-known and highly respected *i*-technology writers and editors.

2 Own a most prestigious blog address!

blog-n-play™ gives you the most prestigious blog address. There is no other blog community in the world that offers such a targeted address, and comes with an instant targeted readership.

3 Best blog engine in the world...

blog-n-play™ is powered by ***Blog-City***™, the most feature rich and bleeding-edge blog engine in the world, designed by Alan Williamson, the legendary editor of ***JDJ***. Alan kept the *i*-technology community bloggers' demanding needs in mind and integrated your blog page to your favorite magazine's Web site.

www.TAMI.linuxworld.com

"Many blogs to choose from"

PICK YOUR MOST PRESTIGIOUS ADDRESS

IT Solutions Guide
Storage+Security Journal
JDJ: Java
Web Services Journal
.NET Dev. Journal
LinuxWorld Magazine
LinuxBusinessWeek
Eclipse Dev. Journal

MX Dev. Journal
ColdFusion Dev. Journal
XML-Journal
Wireless Business &Tech.
WebSphere Journal
WLDJ: WebLogic
PowerBuilder Dev. Journal

3 MINUTE SETUP

Sign up for your FREE blog Today!

blog-n-play.com™
i-Technology Blogs Read by Millions
beta
www.blog-n-play.com
— This site will go beta February 15, 2005

Novell and Red Hat Join IBM Open Source Strategic Alliance Program

Novell and Red Hat have been elevated to IBM's Strategic Alliance program -- IBM's highest tier partner status, according to a recent IBM announcement. The move will make it simpler for clients to acquire open standards-based Linux hardware, software, and services through integrated and streamlined sales, distribution and services channels.

Inclusion in the program opens significant new channels and access to IBM innovation centers to Novell and Red Hat, including those in emerging countries like Brazil, Russia, India, China and Korea, IBM says, "to drive more open source deals in those booming markets."

Other results of the move include new subscription models combining IBM offerings with Novell and Red Hat solutions; support by Novell and Red Hat of IBM's open platforms including the Java-based Apache Geronimo web server and Apache Derby database; and continued support of the Eclipse development platform.

"The companies are (also) reinforcing their commitment to the Java community, which comprises more than six million developers worldwide," IBM said in an official statement. "Additionally, the companies will help customers begin to deploy Service Oriented Architectures (SOA) based on a J2EE application environment."

IBM says this announcement represents the most significant elevation of IBM's strategic partnerships with its key Linux Distribution Partners since it embraced Linux six years ago. The company cited figures from IDC that show that the overall Linux marketplace is growing annually at 26 percent, and expected to reach $35.7 billion by 2008."IBM made a big commitment to our customers and our open source strategy when we told the world we were making a substantial investment in Linux nearly six years ago. That included partnering with Red Hat and Novell," said Mark Elliott, general manager, Global Solution Sales, IBM Sales and Distribution. "Our customers responded with incredible enthusiasm and today IBM's Linux business is growing faster than the industry. And that growth is propelling Novell and Red Hat to the top tier of IBM's strategic alliance partners."

IBM also says it now ranks number one in overall Linux-based server revenue worldwide with 29.7 percent of the revenue, up 32 percent year-to-year, according to Gartner, an industry research firm.

One- and three-year Linux server subscriptions will be available on IBM server hardware, or with IBM middleware, and IBM Services' SupportLine offerings, providing support for the Linux operating system running on both IBM and non-IBM server hardware.

The new subscription model is aimed at benefiting IBM's large enterprise and small and medium business (SMB) clients desiring to purchase operating system subscriptions and support as part of a more comprehensive industry-specific solution from IBM.

Under the terms of the Strategic Alliance agreement, Novell has agreed to distribute the Apache Geronimo open source J2EE application server as part of its SUSE Linux Enterprise Server distribution, which contains more than 1,400 open source software packages. Novell currently includes the Apache Derby database in SUSE Linux 10.0.

The software provides mid-sized businesses, departments in large enterprises and Business Partners with access to open source-based technologies with no up-front costs. Red Hat will work with IBM to certify IBM's version of the Geronimo application server -- WebSphere Community Edition -- for Red Hat offerings and Red Hat will also support IBM's efforts to promote Apache Geronimo.

Novell plans to include Apache Geronimo in its next SUSE Linux Enterprise Server distribution to be delivered in 2006, and Red Hat will support IBM in its efforts to promote Apache Geronimo. Both companies have also agreed to help promote Apache Derby, a Java-based relational database that IBM contributed to the open source community in August 2004 to help developers more easily build and deploy applications and workloads that require an embedded database. Examples of such solutions include small Web sites, point-of-sales systems, local registries and repositories, and small departmental applications.

Covansys Launches IBM WebSphere Concept Center

Covansys has created Concept Center for IBM Websphere DataStage, in partnership with IBM, "to strengthen its offerings in the Business Intelligence Practice with respect to 'Real-Time Information Integration' capabilities," the company says.

Incubated as part of the Business Intelligence Center of Excellence, the center will develop and provide Data Warehousing and Business Intelligence solutions to extend into the Real-Time Information Integration space, which global customers across different industries were demanding, according to a Covansys statement.

The Covansys Concept Center supplements the existing Covansys WebSphere Innovation Center, and will be "a key facility to consult, design and architect Real-Time Information Integration frameworks and advise its customers on best practices for the design, development and deployment of BI frameworks, which were scalable and also flexible to address changing business demands," according to Covansys.

The center will host a lab environment for Covansys' consultants to train on and experience different implementations of DataStage for and on behalf of their customers, providing customers a practical view as they embark on the On Demand journey. Covansys will also work closely with IBM on latest versions and advanced blueprints, validate them against real-life customer scenarios and provide feedback to IBM.

Prolifics Focuses on SOA Web Services and J2EE Migrations to IBM WebSphere

Prolifics has a dedicated practice around technology migrations to IBM WebSphere from other J2EE application servers, such as BEA WebLogic, Oracle Application Server, Sun iPlanet, or JBoss. Prolifics can also adapt applications to support multiple application servers from the same code base., the company says.

Having conducted dozens of

migrations, the company has also built a knowledgebase of compatibility and functional differences, as well as best practices around migration techniques. The company says it leverages its expert consultants with experience, tools and a methodology that avoids unnecessary risk, time, effort and cost.

Also, as one of IBM's highly-specialized WebSphere Portal Service Providers, Prolifics specializes in delivering high-performing and scalable Portal solutions which include:

- Employee portal
- Corporate extranet
- Partner portal
- Customer self service
- Executive dashboards
- Collaborative workspace
- eGovernment portal
- B2B/B2C portals

Prolifics delivers portals with the goal to achieve a unified, personalized and collaborative experience for both employees and customers that streamlines businesses and improves customer and partner experiences. Portals provide a single point of interaction for applications and content, the company points out, as well as a customized and personalized view based on role, single sign-on to all of those applications, collaborative capabilities, workflow automation, sharing of content and documents, and customer self-service capabilities.

Instantiations Leverages IBM WebSphere and Eclipse Alliances

Instantiations leverages its deep proprietary technology base, enterprise software market experience, and world class technical expertise to provide performance enhancement products and services to software professionals who are building, deploying and managing a range of Java systems from stand-alone to fully distributed multi-server e-business applications.

Individually and as a group, the Instantiations team has been responsible for many significant technology firsts and accomplishments: Development of JOVE, the first Optimizing Native Compiler for Java Technology; Development and marketing of the first commercial Smalltalk development environment; Creation of a market dominating line of GUI building tools for Smalltalk; Creation of the first commercial generation scavenging garbage collector; Development of one of the first optimizing compilers for Pascal.

The company's CodePro AnalytiX, for example, with more than 700 rules and metrics, delivers a rich set of continuous collaborative code analysis (C3A) tools that analyze and improve your Java applications. CodePro AnalytiX is easy to install, intuitive, and helps developers improve the quality of their code while increasing productivity, the company says.

CodePro AnalytiX provides features like code audit, metrics, JUint test generation, code coverage, dependencyanalyzer, Javadoc repair and team collaboration. These unique tools automate manual tasks, offer immediate feedback and streamline development activities. What had been time-consuming tasks are, with CodePro AnalytiX, carried out with a few keystrokes, freeing you to spend time on the more creative aspects of software development.

CodePro AnalytiX integrates seamlessly into Rational Developer, IBM WebSphere Studio or any Eclipse development environment. It extends the functionality of the host IDE, resulting in greater capability, and it allows developers to work with the tools of their choice.

Tangosol provides in-memory data management, caching and grid computing solutions for mission critical J2EE applications. The company's Coherence software strives to lead the market in helping organizations achieve extreme performance, continuous availability and unlimited scalability predictably and economically -- even under the most demanding and erratic load conditions, the company says.

The company's recent Tangosol Coherence 3.1 Pre-Release (build 325) download is now posted. This build adds the following new features and enhancements:

- Management Framework Enhancements
  - Ability to register any compliant Standard or Dynamic MBean type with the Management Framework.
  - o A number of additional attributes to the ClusterNode and Service MBeans.
  - o New StorageManager MBean type that surfaces additional management attributes and operations for the storage-enabled portion of a partitioned cache.

- Extend-JMS Enhancements
  - Significantly optimized and improved the performance of a NearCache backed by a JMS NamedCache stub
  - Ability to map a cache name directly to a JMS caching scheme in a cache configuration descriptor file.
  - Support for additional NamedCache operations to the JMS NamedCache stub and to the NearCache when a JMS NamedCache stub is used as the back map.
  - Additional configuration capabilities.

- Web Enhancements
  - New HttpSessionManager MBean for managing and monitoring Coherence*Web installations.
  - Support for JSPs that extend a base Servlet class.
  - Coherence*Web Installer Ant task upgraded to Ant 1.6.5.

- Berkeley DB External Caching Scheme Coherence 3.1 introduces support for Berkeley DB backed overflow maps. When enabled within a Coherence overflow scheme any data overflowing a cache from this scheme will be moved into an on-disk Berkeley DB binary store. This functionality requires the Berkeley DB Java Edition class libraries which may be obtained separately.
- Overflow Caching Enhancements includes dramatically expanded overflow caching functionality
  - OverflowMap now supports the full CacheMap API, including expiry.
  - Support for re-entrancy and caching misses was added.
  - Concurrency (and thus multi-threaded throughput) has been improved.
  - A number of optimizations have been added for high-latency disk stores
  - The memory footprint of the overflow feature has been reduced in general,

and a new overflow option has been added to support overflowing very large data sets to disk without using any additional memory.

ClearStory Systems Announces Availability of Enterprise Media Server (EMS) for IBM WebSphere Application Server

ClearStory Systems has extended support for industry-leading application servers with the availability of Enterprise Media Server (EMS) for IBM WebSphere Application Server.

"Leading companies rely on ClearStory to help them achieve business agility and accelerate time to market through the management of their rich media supply chain," said Hank Nelson, ClearStory president and chief executive officer. "Support for IBM WebSphere expands the application server options available to our customers, allowing them to leverage existing technology investments for a lower total cost of ownership.

Unlike the second-generation DAM products on the market, EMS is a pure Java(TM) 2 Enterprise Edition (J2EE) platform with a service-oriented architecture (SOA) for a peerless third-generation solution for the management of rich media-video, multi-media presentations, graphics, images and publishing documents-the fastest growing content in most enterprises today. EMS provides a scalable enterprise-class platform for developing rich media applications or integrating this content into enterprise business processes.

As the foundation of the IBM WebSphere(R) software platform, WebSphere Application Server, provides a J2EE and Web services offering to deploy, integrate, and manage dynamic e-business applications, with full J2EE V1.4 compatibility, multiple deployment options, improved ease of use and broad cross-platform support. This recent release of EMS allows ClearStory customers to take advantage of the benefits of WebSphere Application Server.

The Enterprise Media Server platform is also certified for the BEA WebLogic and JBoss application servers and supports IBM DB2, Microsoft SQL, and Oracle 9i databases. EMS can be deployed as a standalone product or can be incorporated as part of a comprehensive, enterprise content management strategy.

OpenPages Announces Availability of SOX Express for IBM WebSphere Application Server

OpenPages, the leading provider of enterprise governance, risk and compliance management (GRCM) solutions, today extended support for industry-leading application servers with the availability of SOX Express for the IBM WebSphere Application Server.

"The world's largest corporations rely on OpenPages to reduce the time and cost burdens of Sarbanes-Oxley compliance," said Peter Emerson, vice president of business development at OpenPages. "With support for IBM WebSphere, our customers can deploy SOX Express on their choice of application servers, leveraging existing technology investments and conforming to corporate standards."

As the foundation of the IBM WebSphere software platform, WebSphere Application Server, Version 6.0 provides a Java 2 Enterprise Edition J2EE) and Web services offering to deploy, integrate, and manage dynamic e-business applications, with full J2EE V1.4 compatibility, multiple deployment options, improved ease of use and broad cross-platform support.

OpenPages SOX Express is an enterprise compliance management solution that reduces the time and resource costs associated with ongoing compliance for Sections 302 and 404 of the Sarbanes-Oxley Act. SOX Express combines powerful document management, business process management and flexible reporting capabilities in an extremely easy-to-use environment that enables an organization to document financial controls and automate the ongoing test and review process. Completely web-based, SOX Express maximizes end-user participation with little training. Its dashboards can be used by project managers, documentation team members, internal auditors and external auditors to plan, document and test the internal controls framework of the company, and to attest to the financial statements.

Oracle to Partner With IBM to Further Develop Industry-Leading Applications on AIX Operating Systems

Oracle today announced that the company will join the IBM AIX Collaboration Center (IACC) as a founding partner. As a member of the IACC, Oracle developers will work closely with IBM to more tightly integrate Oracle(R) applications with the current and upcoming versions of AIX 5L, and will continue to deliver Oracle applications on AIX concurrent with their availability on other platforms. Through this collaboration, customers will benefit from greater exploitation of each platform's leadership features including virtualization, security and cross-platform portability. Today, Oracle and IBM share more than 11,000 customers.

"Our applications provide customers with the tools they need to manage the information that runs their businesses," said Charles Phillips, president, Oracle. "IBM has become the leader in the UNIX marketplace and our partnership with the company will allow Oracle to take advantage of the momentum they have generated over the last few years."

IBM System p5 and eServer p5 systems lead the industry with more than 70 key computing performance benchmarks. Earlier this year, Oracle and IBM announced a world record TPC-C benchmark for performance for a 32-processor system.

"Oracle's interest and investment in the AIX development platform is a recognition of IBM's leadership in the Unix marketplace, and our focus on creating technologies that are fundamental to the futures of both our customers and partners," said Adalio Sanchez, General Manager, IBM eServer pSeries.

AIX 5L runs across the entire range of IBM eServer pSeries systems, from entry-level servers and workstations to powerful supercomputers systems, which runs some of the most complex commercial and technical workloads in the world.

eclipse) developer's journal

A DIGITAL PUBLICATION

Your One-Stop Guide to the Eclipse Ecosystem

The main purpose of *Eclipse Developer's Journal* (*EDJ*) is to educate and inform users of Eclipse and developers building plug-ins or Rich Client Platform (RCP) applications. With the help of great columnists, news you can use, and technical articles, *EDJ* is a great information source that you will be able to use to educate yourself on just about anything Eclipse related

WHY 'ECLIPSE DEVELOPER'S JOURNAL' AND WHY NOW? PG 2

eclipse) developer's journal

PREMIER 2005 VOL 1 ISSUE 1

http://eclipse.sys-con.com

What's New in Eclipse 3.1

Polishing the apple ...4

PLUS...

WEB TOOLS PLATFORM: J2EE DEVELOPMENT THE ECLIPSE WAY ...12

'BEING A BETTER PLATFORM ...22
INTERVIEW WITH MIKE MILINKOVICH, EXECUTIVE DIRECTOR, ECLIPSE FOUNDATION

MYECLIPSE 4.0 UP ON DECK: EXCLUSIVE INTERVIEW WITH MAHER MASRI, PRESIDENT, GENUITEC ...36

ECLIPSE IDE FOR STUDENTS: USEFUL TIPS AND TRICKS ...28

Subscribe Today!

FOR YOUR DIGITAL PUBLICATION
(AVAILABLE IN DIGITAL FORMAT ONLY)

6 Issues for $19.99

Visit our site at www.eclipse.sys-con.com or call 1-888-303-5282 and subscribe today!

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

http://eclipse.sys-con.com

SYS-CON.COM

SYS-CON Media, the world's leading publisher of *i*-technology magazines for developers, software architects, and e-commerce professionals, brings you the most comprehensive coverage of Eclipse

WAS and ESB

Connect Non-SOAP HTTP Requesters and Providers to WebSphere Application Server V6 Enterprise Service Bus

BY GREG FLURRY

This article shows you how to connect non-SOAP HTTP service requesters and providers to the IBM WebSphere Application Server V6 Service Integration Bus. This lets requesters and providers leverage the integration capabilities of an enterprise service bus.

Last month's article shows you how to connect non-HTTP service requesters and providers to the IBM WebSphere Application Server V6 Service Integration Bus. This lets requesters and providers leverage the integration capabilities of an enterprise service bus. In this continuation we pick up at Listing 4.

Listing 4 shows the details of the sendJMSMessage() method for the servlet. It creates a JMS bytes message and, if appropriate, inserts the HTTP payload into the body of the bytes message. It then sets the HTTP method in an application-specific JMS property. Next the method inserts any HTTP headers and parameters into the message as application-specific properties in the JMS message. Note that the names used for the JMS properties are derived from the HTTP header or parameter names; the names are given a prefix that should make them unique and distinguish the headers from parameters; also the utility function XMLHTTPUtil.fixName() replaces any dash characters in a name with underscore characters. The method also puts pathInfo from the HTTP request into the JMS message as an application-specific JMS property. Finally the method sends the message to the request queue. Once the message is sent, sendJMSMessage() forms a JMS message selector that tests whether the correlation ID of an incoming response message equals the message ID of the outgoing request message, which supports correlation of the response with the request.

Listing 5 shows the details of the getJMSResponse() method for the servlet. It creates a consumer listening for a message with the selector that correlates the response to the request, as discussed above. The method then places information from the response into an HTTPOutput instance, including the HTTP status code, the response body and any response headers. Note that getJMSResponse() removes the a prefix from the returned HTTP headers.

Listing 6 shows the skeleton for the MDB that forms the off-ramp. When the MDB gets created, the ejbCreate() method creates all of the JMS resources required; in this case, the connection factory, the reply queue and the connection. These exist until the MDB is destroyed. At that time, the ejbRemove() method closes the connection, which releases all the JMS resources.

The ejbCreate() method also determines the root URL used to form the new outbound HTTP request. In the code show in Listing 6, the default value is derived from an EJB environment variable set in the deployment descriptor of the EJB module containing the MDB. The default value could be over-ridden by setting a new JMS property in some mediation that runs in the SIBus; however, overriding is not implemented in this example.

ABOUT THE AUTHORS

Greg Flurry is a Senior Technical Staff Member in IBM's Enterprise Integration Solutions group. His responsibilities include working with customers on service oriented solutions and advancing IBM's service oriented products.

BY NOW THERE ISN'T A SOFTWARE DEVELOPER ON EARTH WHO ISN'T AWARE OF THE COLLECTION OF PROGRAMMING TECHNOLOGIES KNOWN AS AJAX!

REAL-WORLD AJAX
ONE DAY SEMINAR

www.ajaxseminar.com

March 13, 2006

Marriott

Marriott Marquis Times Square
New York City

For more information
Call 201-802-3022 or
email events@sys-con.com

How, in concrete terms, can you take advantage in your own projects of this newly popular way of delivering online content to users without reloading an entire page?

How soon can you be monetizing AJAX?

This "Real-World AJAX" one-day seminar aims to answer these exact questions...

Led by "The Father of AJAX" himself, the charismatic Jesse James Garrett of Adaptive Path, "Real-World AJAX" has one overriding organizing principle: its aim is to make sure that delegates fortunate enough to secure themselves a place – before all seats are sold out – will leave the seminar with a sufficient grasp of Asynchronous JavaScript and XML to enable them to build their first AJAX application on their own when they get back to their offices.

Jeremy Geelan

Jeremy Geelan
Conference Chair, Real-World AJAX
jeremy@sys-con.com

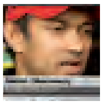
HURRY!
LIMITED SEATING
THIS SEMINAR WILL
SELL-OUT
CALL 201-802-3022
TO REGISTER!

Jesse James Garrett
Father of AJAX

Scott Dietzen
Creator of WebLogic, Ph.D., President and CTO, Zimbra

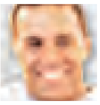
Bill Scott
AJAX Evangelist of Yahoo!

David Heinemeier Hansson
Creator of Ruby on Rails


Satish Dharmaraj
Father of JSP, Co-Founder & CEO, Zimbra

Rob Gonda
Best-Selling AJAX Author, CTO, iChameleon Group

Dion Hinchcliffe
Co-founder & CTO, Sphere of Influence Inc.


Ross Dargahi
Well-known AJAX Evangelist & Co-founder and VP of Engineering, Zimbra

Early Bird (Before January 31, 2006)	$995
Discounted Price (Before February 28, 2006)	$1,195
Seminar Price (After February 28, 2006, and if any seat is available)	$1,295

MEDIA SPONSOR
ajax developer's journal

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

PRODUCED BY SYS-CON EVENTS

COPYRIGHT ©2006 SYS-CON MEDIA ALL RIGHTS RESERVED NOTE: SPEAKER LINE-UP SUBJECT TO CHANGE WITHOUT NOTICE

Listing 7 shows the details of the onMessage() method of the MDB. The method first retrieves the information about the original HTTP request from the incoming JMS message, including the request body, the HTTP parameters and headers, the HTTP "infoPath" and the HTTP method. Note that HTTP headers and parameters are distinguished by the prefix on the JMS property names. The HTTP parameters are added onto the root URL set in the ejbCreate() method. The onMessage() method then calls the private sendHTTPRequest() method to send the new HTTP request and wait for the response.

After receiving the HTTP response, the onMessage() method extracts the HTTP response information and puts it in a JMS message sent to the reply queue. This message includes the response status, any headers and the body, as needed. The headers must have the names converted, as described above. Note in particular that the method sets the correlation ID of the JMS message to the message ID of the incoming request to support correlation of the response with the request in the servlet. The method then sends the JMS message to the reply queue.

Listing 8 shows the details of the sendHTTPRequest() method of the MDB. The method inserts the headers, the body and the HTTP method into the request. The outgoing HTTP request is therefore semantically equivalent to the original HTTP request except for the root URL. The method sends the HTTP request to the target provider and waits for the HTTP response.

Once it receives the HTTP response, the sendHTTPRequest() method extracts the response and puts it into an instance of HTTPOutput. The instance contains the status code, and may also contain headers and a body. Note that if there is an error in the HTTP response from the provider, the attempt to get the HTTP response InputStream causes an exception. In that case, the HTTPOutput instance contains only the HTTP status code.

Configuration

In Figure 4, you can see that two SIBus queue destinations need to be created for the ESB to support XML/HTTP in the manner described in this article. The two queues, one for the request and one for the response, must be exposed as part of a JMS provider. In particular, you need to create a JMS connection factory for your SIBus and a JMS queue for both SIBus queues. Since the MDB shown in Figure 4 listens for messages on the request queue, you must also create a JMS activation specification for the request queue. Building an Enterprise Service Bus with WebSphere Application Server V6, Part 3 describes these steps in more detail. In the sample configuration that corresponds to the sample code described above, the JMS connection factory has a JNDI name of jms/MDBCF, the request JMS queue has a JNDI name of jms/MRequest, the response JMS queue has a JNDI name of jms/MReply, and the JMS activation specification has a JNDI name of eis/TestMDBAS.

The servlet references the JMS resources defined above. The deployment descriptor for the Web module containing the servlet must define resource references for the connection factory and both the request and reply queues. In Rational® Application Developer, you can define the resource references on the References tab of the Web project deployment descriptor editor, by entering the JNDI names: *jms/MDBCF, jms/MRequest and jms/MReply*.

The MDB listens for incoming requests on the request queue and sends responses on the reply queue. To allow the MDB to send the response, the deployment descriptor for the EJB module containing the MDB must define resource references for the connection factory and reply queue. In the Rational Application Developer environment, you can define the resource references for the MDB on the References tab of the EJB project deployment descriptor editor, entering the JNDI names: *jms/MDBCF and jms/MReply*. To listen for the request, the deployment descriptor for the EJB module containing the MDB must define the JMS activation specification and the request queue resources. In Rational Application Developer, you can define the resource references for the MDB on the Bean tab of the EJB project deployment descriptor editor by entering the JNDI names *eis/TestMDBAS* and *jms/MQ* in the WebSphere Bindings area under the JCA Adapter property.

The MDB also uses an environment variable to set the root URL for the outbound HTTP request. The deployment descriptor for the EJB module containing the MDB must define the environment variable. In Rational Application Developer, you can define the environment variable for the MDB on the Bean tab of the EJB project deployment descriptor editor, by entering the appropriate name and value of the environment variable in the Environment Variables area. In our example, the environment variable name is *Default_URI_Root*.

An example

Listing 9 shows a POST request generated by a non-SOAP HTTP requester that happens to use an XML body. Note that the root URL of the service provider is /XMLHTTP/XMLHTTP2SIB. Therefore, there is an HTTP InfoPath that equals *dofuss*. There are also two HTTP parameters and several headers, one specifically introduced by the requester (A-FAKE-HEADER).

Listing 10 shows the request sent to the default target identified in the MDB environment. You can see the new root, /FakeXMLHTTPService/FakeService, defined by the EJB environment variable (which also contains the host name and port information). You can also see the InfoPath and HTTP parameters are included, though the syntax is a bit different, as the parameters are out of order. You can see that all the original headers appear in the new request, but again, the syntax is a bit different in that the headers are in a different order. Finally the body appears in the new request identical to the original body. Thus, even though the new request syntax is a bit different, the semantics of the original request are preserved.

Listing 11 shows the original response to the POST request. Note that it contains a custom header (XYZ-RQF) and a Set-Cookie header as well as an XML body.

Listing 12 shows the response actually returned to the requester. Note that as with the request, the response returned to the requester is semantically equivalent to the original response, even though the syntax is a bit different.

You should recognize that while the mechanisms described above ensure semantic fidelity in the request and response, it is possible that integration logic running in the ESB (for example, mediations running in the SIBus) could intentionally break that fidelity. For example, it might be necessary to add HTTP headers or even transform the body.

Summary

This article showed you how to attach non-SOAP HTTP service requesters and service providers to an ESB built with the WebSphere Application Server V6 SIBus. The detailed example ensures the semantic fidelity of the request and reply to support the full spectrum of non-SOAP HTTP requesters and providers. This allows your ESB to provide integration logic for non-SOAP HTTP requesters and providers, just as it can for SOAP/HTTP and JMS requesters and providers

Resources

Learn

- Building an Enterprise Service Bus with WebSphere Application Server V6, Part 1: Introduction to WebSphere V6 Messaging Resources: Learn how to use the new messaging engine in WebSphere Application Server V6 to build an Enterprise Service Bus.
- Building an Enterprise Service Bus with WebSphere Application Server V6, Part 3: A simple JMS messaging example: Learn how to set up a message-driven bean to listen to messages on a JMS queue and a J2EE client application to send a message to that JMS queue.
- XML-RPC home page: Learn more about XML-RPC.
- Building Web Services the REST Way: Learn more about REST and how to build Web services REST style.

Get products and technologies

- eBay REST API: Get information on the eBay REST API and download samples.
- eBay XMP API: Get information on the eBay XML API and download the schema.
- Using Yahoo! Search: Get information on Yahoo! Search Web services and download the SDK.

Listing 4:
sendJMSMessage() method of XMLHTTP2SIB servlet

```
private String sendJMSMessage(String type,
HttpServletRequest request)
throws IOException {
  String selector = null;
  try {

  MessageProducer producer =

  session.createProducer(requestQueue);

  BytesMessage outMessage = session.create-
BytesMessage();

  // send request body only when actually have
one

  if (type.equals("POST") || type.equals("PUT"))
{

  ServletInputStream sis = request.getInput-
Stream();

  byte[] body = new byte[sis.available()];

  sis.read(body);
          outMessage.writeBytes(body);
        }

  // set the message type per the incoming
request

  outMessage.setStringProperty("SIB_HTTP_
METHOD", type);

  // set the HTTP headers in the message

  Enumeration enum = request.getHeaderNames();

  while (enum.hasMoreElements()) {

  String name = (String)enum.nextElement();

  String transformedName = XMLHTTPUtil.
fixName(name);

  outMessage.setStringProperty(

  "SIB_HTTP_HEAD_"+transformedName,

  request.getHeader(name));

  }

  // set the HTTP parameters on the message

  enum = request.getParameterNames();

  while (enum.hasMoreElements()) {

  String name = (String)enum.nextElement()

  outMessage.setStringProperty(

  "SIB_HTTP_PARM_"+name,

  request.getParameter(name));

  }

  // set the pathInfo on the message

  String pathInfo = request.getPathInfo();

  String URItail = null;

  if (pathInfo != null) URItail =

  request.getPathInfo().substring(1);

  outMessage.setStringProperty("SIB_PathInfo",
URItail);

        // send the message
        producer.send(outMessage);
        selector = "JMSCorrelationID ='" +

  outMessage.getJMSMessageID() + "'";
        producer.close();
  } catch (JMSException e) {

  …
  }
  return selector;
}
```

Listing 5:
getJMSResponse() method of XMLHTTP2SIB servlet

```
private HTTPOutput getJMSResponse (String selec-
tor, HttpServletResponse response){
byte [] realResponse = null;
      HTTPOutput output = null;

  try {

  MessageConsumer consumer =

  session.createConsumer(responseQueue,

  selector);


  Message message = consumer.receive();
        output = new HTTPOutput();

  output.setReturnCode(

  Integer.parseInt(message.getJMSType()));

  BytesMessage theMessage = (BytesMessage) mes-
sage;

  realResponse =

  new byte[(int)theMessage.getBodyLength()];

  theMessage.readBytes(realResponse);

  output.setResponse(realResponse);


  // get headers

  Enumeration enum = message.getPropertyNames();

  Vector header = new Vector();

  output.setHeader(header);

  while (enum.hasMoreElements()){

  String name = (String) enum.nextElement();

  if (name.startsWith("SIB_HTTP_HEAD_")) {

  String[] headerEntry = new String[2];

  header.add(headerEntry);
```

```
   headerEntry[0] = name.substring(14);

   headerEntry[1] =

   message.getStringProperty(name);

   }

   }


   consumer.close();
   } catch (JMSException e) {

   …
   }
   return output;
}
```

Listing 6:

Message Driven Bean skeleton

```
public class XMLHTTPMDB implements javax.ejb.
MessageDrivenBean,
   javax.jms.MessageListener {

   // the connection factory and queue names
   private final static String

   JMSCF_JNDI_NAME = “java:comp/env/MDBCF”;
   private final static String

   JMSQ_JNDI_NAME = “java:comp/env/MReply”;
   // the JMS connection information
   private Destination queue = null;
   private Connection connection = null;
   private Session session = null;

   // the root of the URI used to send the HTTP
request
   private String default_URI_Root = null;

   // getMessageDrivenContext and setMessage-
DrivenContext not shown

   public void ejbCreate() {

   try {

   InitialContext context = new InitialContext();

   default_URI_Root = ((String)

   context.lookup(

   “java:comp/env/Default_URI_Root”));

   ConnectionFactory factory =
(ConnectionFactory)

   context.lookup(JMSCF_JNDI_NAME);

   queue = (Destination) context.lookup(JMSQ_
JNDI_NAME);

   connection = factory.createConnection();

   session = connection.createSession(false,

   Session.AUTO_ACKNOWLEDGE);

   } catch (NamingException e) {

   …
   } catch (JMSException e) {

   …

   }
   }

   public void ejbRemove() {

   try {

   connection.close();

   } catch (JMSException e) {

   …

   }
   }

}
```

Listing 7:

onMessage() method of XMLHTTPMDB MDB

```
public void onMessage(javax.jms.Message msg) {
   StringBuffer url = null;
   byte[] body = null;
   String type = null;
   Vector header = new Vector();
   BytesMessage byteMsg = (BytesMessage) msg;
   try {

   // retrieve the HTTP method, message body

   type = msg.getStringProperty(“SIB_HTTP_
METHOD”);

   body = new byte[(int) byteMsg.getBodyL-
ength()];

   byteMsg.readBytes(body);

   // form new target URL

   url = new StringBuffer(default_URI_Root);

   if (msg.getStringProperty(“SIB_PathInfo”) !=
null)

   url.append(

   ‘/’+msg.getStringProperty(“SIB_PathInfo”));

   boolean haveParm = false;

   Enumeration enum = msg.getPropertyNames();

   // HTTP parameters, headers

   while (enum.hasMoreElements()){

   String name = (String) enum.nextElement();

   if (name.startsWith(“SIB_HTTP_PARM_”)) {

   if (haveParm) {

   url.append(‘&’);

   } else {

   haveParm = true;

   url.append(‘?’);

   }

   url.append(

   name.substring(14)+

   “=”+msg.getStringProperty(name));


   } else if (name.startsWith(“SIB_HTTP_HEAD_”))
{

   String[] headerEntry = new String[2];

   header.add(headerEntry);

   headerEntry[0] = name.substring(14);

   headerEntry[1] = msg.getStringProperty(name);

   }

   }
   } catch (JMSException x) {

   …
   }

   // send the HTTP request
   HTTPOutput theOutput = sendHTTPRequest(

   url.toString(), type, body, header);
   // send back the reply via JMS
   try {

   MessageProducer queueSender =

   session.createProducer(queue);

   BytesMessage outMessage = session.create-
BytesMessage();

   // set the status code

   int status = theOutput.getReturnCode();

   outMessage.setJMSType(Integer.
toString(status));

   // write the body

   if (theOutput.getResponse() != null) {

   outMessage.writeBytes(theOutput.getRe-
sponse());
          }

   // set the correlation ID
          outMessage.setJMSCorrelationID(msg.
getJMSMessageID());
          // insert the response headers
          Vector headerVec = theOutput.getH-
eader();
          if ((headerVec != null) && (headerVec.
size() > 0)) {

   for (int i=0; i < headerVec.size(); i++) {

   String[] entry = (String[]) headerVec.get(i);

   outMessage.setStringProperty(
```

e-BUSINESS SOLUTIONS: TAKING BUSINESS TO THE NET

SPECIAL: PREMIER ISSUE!

WebSphere DEVELOPER'S JOURNAL

The World's Leading Independent WebSphere Developer Resource

PREMIER VOLUME 1 ISSUE 1

INTRODUCTORY CHARTER SUBSCRIPTION SAVE $30 Today! – SEE PAGE 5 FOR DETAILS –

FROM THE EDITOR It's You, Not he Software BY JACK MARTIN – PAGE 3 –

WEBSPHERE CONNECTIONS WebSphere UG's BY MICHAEL MANDE – PAGE 4 –

COMMENTARY Faster, Cheaper, Smarter BY BOB GAULT – PAGE 18 –

FINAL THOUGHTS Networks BY JIM MARTIN – PAGE 38 –

Iterative Development with VisualAge for Java — Visiting and revisiting your code throughout the cycle — BY PRASAD THAMMINENI PAGE 10

Best Practices in WebSphere Application Packaging and Deployment PAGE 6 BY PRASAD THAMMINENI

Iterative Development with VisualAge for Java PAGE 10 Visiting and revisiting your code throughout the cycle BY WAYNE BEATON

Building an Application Server Data Center PAGE 20 Establishing a solid foundation for your new home BY JOE FARSETTA

WebSphere Migration PAGE 28 ...ach for success BY ROBERT COSTELLO

Product Review: IBM WebSphere Studio v3.5 Advanced PAGE 32 REVIEWED BY PAT MARTIN

...ing JDBC Connections PAGE 34 BY ANDY MCCRIGHT

IBM WebSphere Studio v3.5 Advanced
It's You, Not the Software
Iterative Development with VisualAge for Java
Visiting and revisiting your code throughout the cycle
The World's Leading Independent WebSphere Developer Resource

Do You Have Access to the Internet?

Then Subscribe Online and Save 20%

It's that easy

Simply point your mouse to...

http://websphere.sys-con.com

Limited Time Offer!

WebSphere DEVELOPER'S JOURNAL

SYS-CON Media, the world's leading publisher of *i*-technology magazines for developers, software architects, and e-commerce professionals, brings you the most comprehensive coverage of WebSphere.

SYS-CON MEDIA

SUBSCRIBE NOW AND SAVE 20% OFF THE ANNUAL NEWSSTAND RATE

ONLY $89.99 FOR 1 YEAR (12 ISSUES)

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

```
   “SIB_HTTP_HEAD_”+

   XMLHTTPUtil.fixName(entry[0]), entry[1]);

   }

   }

   // send the response message

   queueSender.send(outMessage);
   } catch (JMSException e) {

   e.printStackTrace();
   }
}
```

Listing 8:

```
sendHTTPRequest method of XMLHTTPMDB MDB
private HTTPOutput sendHTTPRequest(String outboundURL, String method,
byte[] body, Vector header){
   URLConnection conn = null;
   HttpURLConnection hConn = null;
   OutputStream os = null;
   URL url = null;
   HTTPOutput output = new HTTPOutput();

   // send the HTTP request
   try {

   url = new URL(outboundURL);

   conn = url.openConnection();

   conn.setDoInput(true);

   // put the headers in the request

   for (int i=0; i < header.size(); i++) {

   String[] entry = (String[]) header.get(i);

   conn.setRequestProperty(

   XMLHTTPUtil.unfixName(entry[0]), entry[1]);


   }



   hConn = (HttpURLConnection) conn;

   System.out.println(“method: “+hConn.getRequestMethod());

   hConn.setRequestMethod(method);


   // ensure that send body only if needed

   if (method.equals(“POST”) || method.
equals(“PUT”)) {

   conn.setDoOutput(true);

   os = hConn.getOutputStream();

   if (body != null) os.write(body);

   }

   } catch (IOException ex) {

   ...
   }

   // retrieve HTTP response
   int returnCode = 999;
   try {

   // get the content and status

   InputStream is = conn.getInputStream();

   returnCode = hConn.getResponseCode();

   int avail = is.available();

   byte[] response = new byte[avail];

   is.read(response,0,avail);

   output.setResponse(response);


   // get the headers

   Map headerMap = conn.getHeaderFields();

   Set headSet = headerMap.keySet();

   Object[] headName = headSet.toArray();

   Vector headerVec = new Vector();

   output.setHeader(headerVec);

   for (int i=0; i < headName.length; i++) {

   String name = (String)headName[i];

   if (name != null) {

   String[] entry = new String[2];

   headerVec.add(entry);

   entry[0] = name;

   entry[1] = conn.getHeaderField(name);

   }

   }

   is.close();
   } catch (IOException ex) {

   try { // get the return code when no
InputStream

   returnCode = hConn.getResponseCode();

   } catch (IOException ex2) {

   ...

   }
   }
   output.setReturnCode(returnCode);
   return output;
}
```

Listing 9:

```
Original POST request
POST /XMLHTTP/XMLHTTP2SIB/
dofuss?xyz=nub&fuzz=bust HTTP/1.1
Content-Type: text/xml; charset=utf-8
Accept: text/*
Cache-Control: no-cache
Pragma: no-cache
A-FAKE-HEADER: WHATEVER-YOU-WANT
User-Agent: Java/1.4.2
Host: localhost
Connection: keep-alive
Content-Length: 179

<?xml version=”1.0” encoding=”UTF-8”?>
   <GetAccountPositions xmlns=”http://test.com/
GetAccountPositions”>
      <requestData>
         <pdw8_id>10221477</pdw8_id>
      </requestData>
   </GetAccountPositions>
```

Listing 10:

```
New POST request sent to service provider
POST /FakeXMLHTTPService/FakeService/
dofuss?fuzz=bust&xyz=nub HTTP/1.1
Host: localhost
Connection: keep-alive
Content-Length: 179
Content-Type: text/xml; charset=utf-8
User-Agent: Java/1.4.2
Accept: text/*
Cache-Control: no-cache
A-FAKE-HEADER: WHATEVER-YOU-WANT
Pragma: no-cache

<?xml version=”1.0” encoding=”UTF-8”?>
   <GetAccountPositions xmlns=”http://test.com/
GetAccountPositions”>
      <requestData>
         <pdw8_id>10221477</pdw8_id>
      </requestData>
   </GetAccountPositions>
```

Listing 11:

```
Original response
HTTP/1.1 200 OK
Content-Language: en-US
Set-Cookie: my-cookie=my value
Transfer-Encoding: chunked
Date: Fri, 30 Sep 2005 21:47:43 GMT
XYZ-RQF: zipper

<?xml version=”1.0” encoding=”UTF-8”?>
   <AccountPositions xmlns=”http://test.com/
AccountPositions”>
      <reponseData>
         <pdw8_id>10221477</pdw8_id>
      </reponseData>
   </AccountPositions>
```

Listing 12:

```
Response returned to requester
HTTP/1.1 200 OK
Set-Cookie: my-cookie=my value
Transfer-Encoding: chunked
Date: Fri, 30 Sep 2005 21:47:43 GMT
Content-Language: en-US
XYZ-RQF: zipper

<?xml version=”1.0” encoding=”UTF-8”?>
   <AccountPositions xmlns=”http://test.com/
AccountPositions”>
      <reponseData>
         <pdw8_id>10221477</pdw8_id>
      </reponseData>
   </AccountPositions>
```

Subscribe Today!

– INCLUDES –
FREE DIGITAL EDITION!
(WITH PAID SUBSCRIPTION)
GET YOUR ACCESS CODE INSTANTLY!

The major infosecurity issues of the day... ***identity theft, cyber-terrorism, encryption, perimeter defense, and more*** *come to the forefront in* **ISSJ** *the storage and security magazine targeted at IT professionals, managers, and decision makers*

Editorial Mission

Greater Collaboration and Efficiency Through Education

✓ ***ISSJ*'s editorial mission is to showcase proven solutions** that will guide, motivate, and inspire senior IT and business management leaders in the planning, development, deployment, and management of successful enterprise-wide security and storage solutions.

✓ ***ISSJ*** brings together all key elements of data storage and protection, and presents compelling insight into the benefits, efficiencies, and effectiveness gained by focusing on these two critical areas of IT simultaneously.

✓ ***ISSJ*** is an objective, critical source of information that helps storage and security managers make informed management decisions about what is working today, and what they need to plan for tomorrow, and is the only publication that focuses exclusively on the needs of IT professionals who are driving the enterprise storage architecture/ infrastructure while pursuing and incorporating the latest security technologies.

✓ ***ISSJ*** achieves our mission by delivering in-depth features, practical "how-to" solutions, authoritative commentary, hard-hitting product reviews, comprehensive real-world case studies, and successful business models, written by and for professional IT storage and security practitioners.

SAVE 50% OFF!
(REGULAR NEWSSTAND PRICE)

Only $39.99 ONE YEAR 12 ISSUES

www.ISSJournal.com
or
1-888-303-5282

SYS-CON MEDIA
The World's Leading i-Technology Publisher

Extending JAAS

Advanced Authentication in WAS PART 4

BY KEYS BOTZUM, BILL HINES, PAUL ILECHKO, MESSAOUD BENANTAR

The advanced authentication features in IBM WebSphere Application Server V6 support a more flexible authentication model with a new highly customizable authentication framework that's based on – and extends – Java Authentication and Authorization Service (JAAS).

ABOUT THE AUTHORS

Keys Botzum is a Senior Technical Staff Member with IBM Software Services for WebSphere. He holds a Masters degree in Computer Science from Stanford University and a B.S. in Applied Mathematics/Computer Science from Carnegie Mellon University.

Bill Hines is a Senior Certified Consulting IT Specialist with IBM Software Services for WebSphere. His expertise includes installation, configuration, tuning, dynacache, security, troubleshooting, and design/architecture of enterprise J2EE WebSphere applications.

Paul Ilechko is a Senior Solutions Architect with IBM Software Services for WebSphere. He has been involved with WebSphere and J2EE technology almost since their inception. He has a B.Sc. in Mathematics from the University of London.

Messaoud Benantar is a senior software engineer with IBM Software Services for WebSphere. He spent most of his career as a software developer working in IBM Large Systems Division, Software Group/Tivoli and IBM Global Services. His focus is on middleware and computing security.

This is the fourth article in the "Advanced Authentication in WAS" series. We began by taking a look at the advanced authentication features in IBM WebSphere Server V6 that support a more flexible authentication model with a new highly customizable authentication framework that's based on a Java Authentication and Authorization Service (JAAS). In this article we will pick up the discussion in the subject of propagation.

Although not covered in this article, it is possible to develop your own tokens based upon one of these four types, enabling you to control how information is actually propagated from server to server. It is even possible to customize the network representation of the pre-defined WebSphere Application Server tokens. Refer to the WebSphere Application Server Information Center for details.

Horizontal propagationHorizontal propagation is about the ability of credentials (that is, a subject) created on one server to be made available when the same client accesses another server, without passing through an intermediary server (for example, Web authentication).

As we have seen already, a subject is created when a user signs on to WebSphere Application Server. For Web clients, an SSO token is also created and sent back to the browser. This token does not contain the user identity in full; it contains enough information to uniquely identify - and thus retrieve or recreate - a user's subject. The cache key we mentioned earlier is part of the SSO token. When the subject is created, it is placed in the application server's security cache. It can also, for a clustered environment, be configured such that the tokens representing the subject are stored in DynaCache and replicated. In this scenario, if the user accesses a different application server in the cluster, the set of tokens that make up the subject can be retrieved using the SSO token as a key, and then the custom subject can be recreated.

The SSO token contains a unique ID and timestamp that represents the user, optional custom key information, and JMX admin endpoint information for the application server that created (and cached) the subject. When the user accesses an application server with an SSO token, the following steps are taken to retrieve the subject:

1. The local security cache is searched for an existing, instantiated subject. If the subject is found in the cache, it is associated with the current thread and then used for security purposes.
2. If the subject is not found in the local cache, DynaCache is searched for the token set that can be used to recreate the Subject. If the tokens are retrieved from DynaCache, a propagation login is performed.
3. If the DynaCache lookup is not successful, then the JMX endpoint of the originating server is accessed via JMX and the token set is requested. If the tokens are retrieved via JMX, a propagation login is performed.

Figure 20 illustrates this process.

As you can see from the diagram, there are various paths that can be taken depending on the specific situation. As mentioned earlier, there are three distinct login scenarios: initial login, initial login with SSO token, and propagation login. Some of the subtleties associated with the login and propagation processing will be discussed later.

DynaCache usageWebSphere Application Server creates a private security cache in DynaCache. Tokens representing the subject are placed in the cache by the application server and replicated to the replication domain. This is typically a single WebSphere Application Server cluster within a single cell, so when a Web client accesses a different cluster or cell, the subject will not be available via DynaCache and the system will fallback to JMX access (described next). When tokens are obtained from DynaCache and used to recreate the subject, this results in a propagation login, which works rather differently from an initial login, as shown in Figure 21.

JMX usageWhen the subject information is not available from either the local application server security cache or from DynaCache, the runtime will attempt to retrieve the tokens by making a secure JMX admin call to the server that performed the original initial login. Using tokens retrieved by JMX also results in a propagation login. It is possible that the JMX call will fail, either because the originating server is not currently available, or because the credentials are no longer available in the security cache. In this case, the currently accessed server will perform an initial login with the SSO token. The specific behavior in this situation will be described later.

Be aware that if the JMX call crosses cell boundaries, the two cells must share a common security infrastructure -- the same registry, the same LTPA encryption keys, compatible SSL keys, and so on. It is also necessary that the calling application server's identity (the cell security server ID) has administrative authority on the server it is contacting. Only a process with administrative authority can obtain security tokens from another process.

Propagation login vs. initial loginFigure 21 shows the different flow of control between an initial login and a propagation login. Basically, the difference is that a propagation login merely instantiates a Subject using existing information (the tokens), and does not actually authenticate using a registry. On the other hand, an initial login is expected to create the user subject from the available information (perhaps as little as the SSO token).

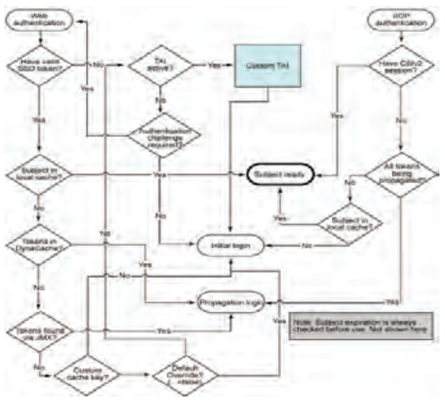


Figure 20: Authentication state diagram

Downstream propagationDownstream propagation is very different from horizontal propagation. In downstream propagation, an application server that has access to the complete user's subject is calling another application server. There is no need here for the sharing features of an SSO token or complex horizontal subject propagation.

- When an application server calls a second server, some information needs to be sent along with the request to continue identifying the user for access to the next resource. The information sent depends upon the J2EE RunAs mode (that is, client, server, or specified). The most common RunAs mode is client, which preserves the originating client identity. When RunAs client is specified in the application's deployment descriptor, the authenticated user subject is set as the invocation subject on the thread ready to be used for any outbound request.
- As with Web propagation, the subject is converted to tokens and these tokens are propagated to the downstream server.
- Upon receipt, the tokens are converted back into a subject via a propagation login. This handshaking occurs as part of the CSIv2 context negotiation. The resulting subject is associated with the CSIv2 session.
- Once a CSIv2 session is established between two parties (including an EJB client), the information in it will remain cached for an extended period. Generally speaking, the session will remain as long as the credentials are still valid. Thus, for example, if an EJB client connects to an application server and remains connected while performing various EJB operations, a custom login module will be called when the client connects and then not again until the client's authentication token expires (two hours by default).

Downstream propagation uses the RMI_INBOUND and RMI_OUTBOUND login configurations. RMI_OUTBOUND is used on the sending server and RMI_INBOUND on the receiving server, as shown earlier.

You can customize what is sent by the upstream server by adding your own login modules to the RMI_OUTBOUND login configuration. For example, you might want to perform some kind of identity mapping at this point.

Client differencesA standalone J2EE client can perform a JAAS login and then make method calls on EJBs in an application server. When this occurs, a subject is created on the server. This can be a custom subject created using a custom JAAS login module. However, there are some issues. First, since there is no SSO token on the client in this scenario, and the custom subject is on the server, not the client, there is no way to perform a propagation login if a client accesses another server (either on failover, or just using another server), and a full initial login will take place at the time the new CSIv2 session is created.

Second, since a full initial login occurs on every access to a new server, there can be problems with repeating the authentication. For example, it is possible that the authentication data is no longer valid, perhaps because the user authentication password is time sensitive.

If a custom Subject is created at the client, it will be propagated to each server instance as part of the initial login, but if it was created on the server by login modules, it must be recreated at the new initial login. Unfortunately, it is not appropriate to store security information at the client since cli-

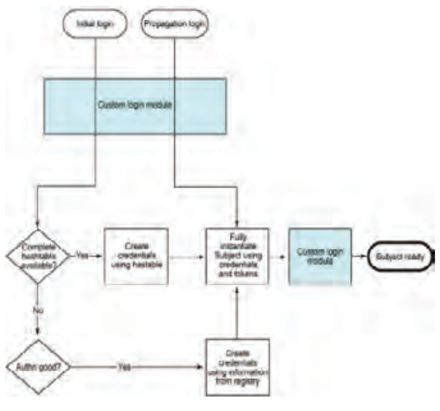

Initial login
Propagation login
Custom login module
Complete hashtable available?
Yes
No
Create credentials using hashtable
Fully instantiate Subject using credentials and tokens
Custom login module
Subject ready
Authn good?
Yes
Create credentials using information from registry

Figure 21: Login types

ents are subject to compromise and generally outside the trust domain. As a result, if you are using Java clients with WebSphere Application Server, you need to think carefully about your design if custom subjects are to be used, because it will be very difficult to share them.

Go back and try the Java client example again. Change the client to connect to two application servers. Notice that the custom login module is called the first time each application server is contacted and the custom subject is recreated.

Propagation: The big picture in Figure 22 shows various aspects of propagation. You will see that attributes are propagated horizontally (for Web SSO) using DynaCache, while attributes are propagated downstream (for RMI requests) using RMI. This is because WebSphere Application Server can propagate attributes directly as part of the CSIv2 protocol for RMI/IIOP requests, only the SSO token is provided to other servers by the Web client for Web requests. Notice also that there is a fallback mechanism in the Web layer. In the event that a Web SSO request arrives at an application server that is not part of the DynaCache replication domain (perhaps it is in another cluster or even another cell), a JMX call is made to the originating application server to obtain the custom subject information. Fallback is not needed for RMI propagation since the calling server simply sends the tokens on the request.

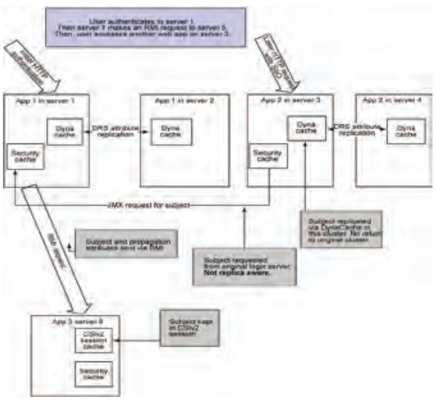

User authenticates to server 1.
Then server 1 makes an RMI request to server 5.
Then, user accesses another web app on server 3.
App 1 in server 1
Dyna cache
Security cache
DRS attribute replication
App 1 in server 2
Dyna cache
App 2 in server 3
Dyna cache
Security cache
DRS attribute replication
App 2 in server 4
Dyna cache
JMX request for subject
Subject replicated via DynaCache in this cluster. No return to original cluster.
Subject and propagation attributes sent via RMI
Subject requested from original login server. Not replica aware.
App 3 server 5
CSIv2 session cache
Security cache
Subject kept in CSIv2 session

Figure 22: Propagation

Issues with horizontal propagation- Propagation is an extremely complex issue, with many subtle complexities you need to be aware of. As discussed earlier, there are several places in which you as a developer have the ability to customize how WebSphere Application Server authentication works. You can add a custom TAI or JAAS login module, you can add custom attributes to the subject, or even alter the way in which WebSphere Application Server creates credentials, thus bypassing the normal registry lookups. When you do this, you need to be very conscious of the issues that can occur, particularly with what happens in a multi-server environment, and how propagation and failover work. In this section, we will try to address some of these issues and point out some good practices. When you perform an initial login, you can customize the subject in either a TAI or a login module. The subject will be created in the application server that you initially access and stored locally in the security cache. It will also be added in token form to a private cache managed by DynaCache, and thus potentially made available to all servers in the same replication domain as the original server. If the subject is a standard WebSphere Application Server subject with no modifications, then the subject can be stored using the default cache key, which is guaranteed to be unique for a given user. However, if you have modified the subject, then you will need to augment the cache key with data that is sufficiently and appropriately unique for the situation. Otherwise, if WebSphere Application Server can't find your cached subject for whatever reason, it will simply recreate a default subject when you hit a different application server.

As far as "appropriate uniqueness" is concerned, this is application specific. If the custom subject is always recreated in the same way, then a simple string including the user's unique ID and some additional string constant is sufficient. However, if two logins for the same user create different subjects (because the content of the Subject varies by, for example, the time of day at login, or location of login device), then the cache key would need to contain sufficient information to ensure different cache keys for different subjects for the same user. For example, the key may contain a hash based on the time, or on the IP address of the login device. The unique cache key information is placed by WebSphere Application Server within the SSO token. Remember that the SSO token is the one and only piece of information that is guaranteed to be shared between application servers when performing Web-based SSO.

When you use a Web browser to access a server other than the one to which you originally authenticated, WebSphere Application Server will attempt to locate the subject as previously described, looking first in the local cache, then in DynaCache, and finally attempting to retrieve the required tokens via a JMX call to the originating server. If the tokens required to recreate the Subject cannot be located by these methods, then things get tricky:

- If there is no custom cache key, WebSphere Application Server will simply perform an initial login using the SSO token as the user's authentication data. The login modules are responsible for recreating the user's subject given only that SSO token. This is exactly the same as the WebSphere Application Server behavior when propagation is not enabled.
- If there is a custom cache key, then by default, WebSphere Application Server will attempt to re-authenticate the user. That is, the SSO token will be ignored and a normal "authenticate from scratch" will be performed. That means that TAIs will be invoked if available, the user will be challenged to authenticate if needed, and then an initial login will be performed. This may be frustrating to your users.
- If there is a custom cache key, but the default behavior has been overridden by setting com.ibm.ws.security.webChallengeIfCustomSubjectNotFound to false for the security subsystem (set via the Global Security Additional properties panel), WebSphere Application Server will attempt to carry on without re-authenticating the user. Instead, an initial login will be performed using only the SSO token. Your custom login modules, if any, will be invoked.

If you override the default behavior by

The World's Leading Java Resource Is Just a >Click< Away!

JDJ is the world's premier independent, vendor-neutral print resource for the ever-expanding international community of Internet technology professionals who use Java.

SYS-CON MEDIA

ONLY $69^99

ONE YEAR
12 ISSUES

Subscription Price Includes FREE JDJ Digital Edition!

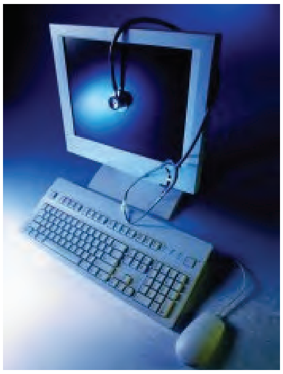
OUT OF MEMORY! PAGE 36
Magazine in the World
JDJ
IN THIS ISSUE...
Leveraging Open Source Solutions to Improve Productivity While Using EJBs PAGE 8
Spring Web Flow PAGE 14
Intelligent Web Applications with AJAX PAGE 28
Navigating the Global Enterprise
JDJ.SYS-CON.COM
ARE PORTALS THE MAGIC BULLET OF WEB APPLICATION DEVELOPMENT?
The many advantages to utilizing portal software
PLUS...
The Performance of EJB 3.0
Dealing with Open Source Software
A Blueprint for Developing Language Tools
THE WORLD'S LEADING i-TECHNOLOGY MAGAZINE JDJ.SYS-CON.COM

www.JDJ.SYS-CON.com

or 1-888-303-5282

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

setting the com.ibm.ws.security.webChallengeIfCustomSubjectNotFound property, when the subject is lost and an initial login is performed, all that is available to the login modules at this point is the SSO token (and the cache key which is contained in the SSO token). If this provides sufficient information for the login module to correctly create the subject, then it can continue; otherwise it needs to terminate with an error, which will force the authentication to fail. Note that at this point you are stuck: you cannot login successfully, but WebSphere Application Server will not attempt to re-authenticate.
The only way to force re-authentication is to destroy the SSO token by shutting down the browser. You should not allow this situation to occur. If you cannot guarantee that your login module can successfully work with only an SSO token, take the default approach of letting WebSphere Application Server repeat the authentication process. This is why the default behavior is to re-authenticate. Only change this if you are sure your login modules can recover.



Notice how the login modules have more responsibility and opportunity for handling complex login scenarios than does a TAI. This is not accidental: login modules provide added flexibility, but at a price. When you write login modules, you are actively integrating with the WebSphere Application Server security runtime, and need to fully comprehend the complexity involved in doing so. The TAI has been kept intentionally simple, and is generally a preferred customer interface in cases where it be used.

Cache issues

- Clearing caches –The lifetime for cached tokens placed in DynaCache is the same as that of the SSO token. Thus, the cached subject data will expire out of DynaCache when the corresponding SSO token expires. The JMX SecurityAdmin clearAuthCache call (see the WebSphere Application Server Information Center for details) that invalidates the security cache entry for a user also invalidates the user's subjects that are cached in DynaCache in addition to the authentication cache. However, the JMX clearAuthCache call does not invalidate the CSIv2 session cache. Therefore, using clearAuthCache will not force the user's subject to be re-computed when the client is using IIOP, unlike with Web clients. Of course, logging in again will result in a new subject being created since the old one is gone.
- Authentication cache implications

Be aware that WebSphere Application Server caches subjects in an authentication cache, which is accessible via many different lookup keys. Some example lookup keys include user ID, user ID and password, and LTPA token. When first authenticating a user, WebSphere Application Server will usually first attempt to find a cached Subject for the user (refer to the earlier state diagram), rather than creating a new subject. This improves performance but may yield unexpected results when using custom subjects.

One key example of this behavior is with EJB client access: If the same user authenticates multiple times with the same user ID and password, the same cached subject will be used, even if the subject is customized on the server side. After the subject is created the first time, your login modules will not be called until the authentication cache expires because the only key WebSphere Application Server has for looking up the potentially cached subject is the user ID and password. You can prevent this behavior by customizing the subject on the client side, just by placing any custom value in it. WebSphere Application Server will then bypass the authentication cache and use the login modules to create the subject.
Go back and try the Java client again. This time, connect repeatedly to the same application server as the same user. Notice that your custom login module is only called the first time. It won't be called again until the authentication cache expires (10 minutes by default).

A second example of this behavior was described earlier in the TAI section (LINK). If a TAI returns just a user ID in the TAIResult object, WebSphere Application Server will look for a subject matching that user ID. This can result in WebSphere Application Server using a previously created and cached custom subject, rather than a default subject. This is why our TAI always creates a subject. In this case, we would prefer that WebSphere Application Server not cache the custom subject based on the user ID (it should use the custom cache key). This behavior means that if any TAI ever creates a custom subject, then every TAI (and login module) must consistently use custom cache keys. This is WebSphere Application Server internal defect #293814 (a fix is not currently available for any release). If this defect impacts you directly, consider contacting IBM Support and requesting a fix.

Enabling propagationIn WebSphere Application Server V6, propagation is enabled by default, so no further action is necessary. You can of course disable propagation if you like. In WebSphere Application Server V5.1.1, propagation is disabled by default. Propagation settings are controlled in two places:

- The LTPA SSO configuration panel enables you to configure Web inbound security attribute propagation.
- The CSIv2 Inbound and Outbound Authentication panels let you configure downstream security attribute propagation.

Conclusion

This article described the advanced authentication features available in WebSphere Application Server to support a more flexible authentication model, including the use of the new, enhanced TAI interface, as well as the custom JAAS login modules. Also discussed were some of the dangers that can trap the unwary developer when dealing with login propagation in a clustered environment.

(A code sample for download can be found by locating this article at IBM developerWorks.)

Welcome to the Future of Video on the Web!

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

> "Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

Welcome to the Future!

Did you already purchase your ".tv" domain name? You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY
SYS-CON EVENTS

List of Topics:

- > Advertising Models for Video-on-demand (VOD)
- > Internet TV Commercials
- > Mastering Adobe Flash Video
- > How to Harness Open Media Formats (DVB, etc)
- > Multicasting
- > Extending Internet TV to Windows CE-based Devices
- > Live Polling During Webcasts
- > Video Press Releases
- > Pay-Per-View
- > Screencasting
- > Video Search & Search Optimization
- > Syndication of Video Assets
- > V-Blogs & Videoblogging
- > Choosing Your PVR
- > Product Placement in Video Content
- > UK Perspective: BBC's "Dirac Project"
- > Case Study: SuperSun, Hong Kong

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4 `	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

COPYRIGHT ©2006 SYS-CON MEDIA ALL RIGHTS RESERVED

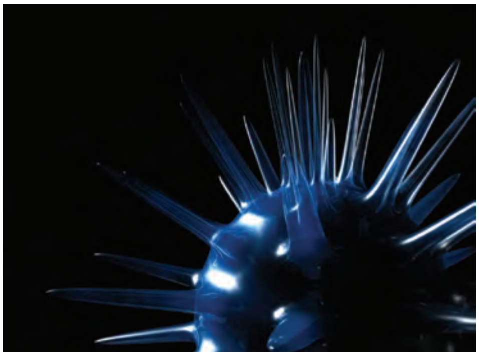
Can We Ever Have Finality?

BY ROGER STRUKHOFF

I've never fully embraced the name of this column, "Final Thoughts." I get it, this is the last page of the magazine, so is the final word in the literal sense. But the column name sounds like a last will and testament, or less ominously, as if this is all the columnist will ever have to say on a particular topic. Is anything final in life, except life itself?

Certainly not in the world of application development, deployment, and management. One of the most conspicuous aspects of today's IT environment, in fact, is the fact that applications are never done, and they must change in a truly fluid, 24/7 environment. No more development "cycles" or scheduled upgrades, or anything that is linked to something as medieval as a monthly and daily calendar.

After a calendar in most of the Western world has days named after ancient Norse gods like Thor, Tiu, and Woden, slightly less ancient Roman rulers, and the very, very ancient Sun and Moon. But even in more practical Japan, with its business-like "one-day," "two-day", etc. naming convention, the calendar is a hopelessly anachronistic method for allowing everyone to get a grip on the task at hand.



There can be no final thoughts in IT development and management today. There is no end of it. And that's the good news.

Business phrases like "maintaining competitive advantage," "enriching the customer experience," or "reducing total cost of ownership" actually have real thoughts behind them. They all strike to the heart of the ultimate goal in a capitalistic society to take more money in than goes out, while competing against all comers in an environment that is not a zero-sum game.

This last aspect is often ignored by dour managers who view any competitor's sale as money lost, but is a critical aspect to the belief that efficiency can drive an economy, and a good economy can drive more efficiency. If the magnificent gains in agricultural productivity over the past 50 years could be matched by similar efficiencies in the way the world consumes and uses energy resources over the next 50 years, then we would truly live in a zero-sum game world that can eliminate poverty, disease, and human suffering.

Oh, and while outlining these utopian dreams, it's worth noting that politicians need to behave a little better as well, whether they are now-brutish dictators in the unfortunate corners of the world or craven demagogues in the more fortunate societies. Even the most sophisticated IT deployments can do little to solve this problem, although the idea of increasingly transparent e-government services does hold promise in delivering government services more efficiently and perhaps even leading to a demand by the world's citizens to expect more transparency from its leaders.

In any case, there will not be a final thought on earth until the day the sun finally morphs into a red giant and undoes all of the earth's eternal verities, from Egypt's pyramids to China's Great Wall to America's Dick Clark.

The subject of the day may change from abstractions such as web services to general approaches such as service-orieneted architectures, to specific products and environments such as WAS in particular and WebSphere in general. But the challenge never changes, and that is to realize that everything has not already been invented, as many thought at the end of the 19th century, that we are nowhere near the end of history, as at least one misguided academic thought at the end of the 20th century, and that the world is not flat, as one misguided author and many misguided commentators seem to think at the dawn of the 21st century.

Because at the end of the day, all we can do is push things forward incrementally, mindful of the ancient calendar perhaps, but aware that customer needs will continually evolve, as will therefore the necessity to always think of what is the next thing that needs to be done. And that's my final thought on this subject…for now.

ABOUT THE AUTHOR

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.
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.

The Developer Paradox:

No time to test your code? But long hours reworking it & resolving errors?

Check out Parasoft Jtest® 7.0

Automates Java testing and code analysis.
Lets you get your time back and deliver quality code with less effort.

- **Automated:**
  Automatically analyzes your code, applying over 500+ industry standard Java coding best practices that identify code constructs affecting performance, reliability and security.

  Automatically generates and executes JUnit test cases, exposing unexpected exceptions, boundary condition errors and memory leaks and evaluating your code's behavior.

  Groundbreaking test case "sniffer" automatically generates functional unit test cases by monitoring a running application and creating a full suite of test cases that serve as a "functional snapshot" against which new code changes can be tested.

- **Extendable:**
  Industry standard JUnit test case output make test cases portable, extendable and reusable.

  Graphical test case and object editors allow test cases to be easily extended to increase coverage or create custom test cases for verification of specific functionality.

- **Integrated:**
  Integrates seamlessly into development IDE's to support "test as you go" development, and ties into source control and build processes for full team development support.

To learn more about Parasoft Jtest or try it out, go to www.parasoft.com/JDJmagazine

PARASOFT®
We make software work.™

Automated Software Error Prevention™

For more information, please call 888-305-0041 (select option "1"). ©2005 Parasoft Corporation. All other company and/or product names mentioned are trademarks of their respective owners.