WebSphere®

The World's Leading Independent WebSphere Developer Resource

DEVELOPER'S JOURNAL

WebSphereDevelopersJournal.com

AUGUST VOLUME 1 ISSUE 7

ANNOUNCING
WEB SERVICES EDGE
WEST COAST!
OCT. 1–3, 2002
SAN JOSE, CA

web services conference&expo **EDGE**

SAN JOSE MCENERY CONVETION CENTER

SEE PAGE 23 FOR DETAILS

FOCUS: Life Sciences

Technology is changing drug research and healthcare

DISPLAY UNTIL OCTOBER 31, 2002
$15.00US $16.00CAN

08>

0 71486 03422 3

SYS·CON MEDIA

PROLIFICS

WWW.PROLIFICS.COM

WILY TECHNOLOGY

WWW.WILYTECH.COM

Awash in Red Ink

BY JACK MARTIN

The U.S. equity markets are awash in red ink. Every day a new financial scandal is announced or some company misses their numbers. Nervous speculators crush the stock price instantly, sometimes taking an entire sector with them.

What caused this is that the core of the financial markets is based on the truth of the financial performance reported by each company and certified by an outside auditor, which is then compared to other investment alternatives. Investors are supposed to choose based on perceived risk and hoped-for reward. Securities are priced in an open auction market, with each competing for investor dollars. What happens in practice is very different.

First, the investing public is to blame for buying stocks in companies without the faintest idea of what the company does to make profits or the competitive environment. This is the cause of the inflated stock market and its subsequent decline.

Most who buy stock approach the purchase with the same care and expectations they have in buying a lottery ticket – unlimited upside and limited downside. When buying a lottery ticket the expectation is that a random turn of the wheel can deliver a windfall or the ticket buyer loses his dollar.

Others buy stocks based on a hot tip from a friend, the media, or even worse, a financial planner. The hot tip is never investigated with due diligence. Instead, an irrational decision is based on the unsubstantiated rumor. The investor rarely acknowledges that the friend knows little about investing, the media person is an actor reading a script, and the adviser may have less money than they do.

Think about it. How many of your friends, the people you watch on television, or investment advisers directly available to you have made a substantial fortune over the long term investing in Wall Street? No one – unless you know Warren Buffet or Charles Munger of Berkshire Hathaway, there is no public record in the history of Wall Street of any individual deriving the vast majority of their wealth buying stocks as an investment.

The rich made their money somewhere else. For example, almost all of Bill Gates' net worth is derived from promoting and selling Microsoft stock, not buying it. The Walton family has also done well promoting and selling Wal-Mart stock, as have many others.

Let me give you an idea of what can happen when a person who made their fortune selling stock to the public goes out and starts buying up stock. Look at William C. Durant, the president and founder of General Motors; he finished penniless as a short-order cook in a luncheonette in Detroit.

The public doesn't seem to understand that you make money on a stock by selling it, not buying it. In the world of auditing, obvious truths are routinely ignored. Independent auditors should be independent – otherwise, there's no point in having them. Companies employ internal accountants to produce the numbers released to shareholders. The auditors' only contribution is an outside certification that the numbers are true. It doesn't take an MBA to figure out that if you give your auditors tens of millions of dollars in consulting fees, they've probably already been corrupted, or at best compromised.

The Securities and Exchange Commission has proposed requiring a company's principal executive officer and principal financial officer to certify that to their knowledge, the information in the company's quarterly and annual reports is true in all important respects. And, that the reports contain all information about the company of which they are aware and believe important to a reasonable investor.

WorldCom's bankruptcy will probably force other U.S.-based telecoms into bankruptcy as well. Filing will give WorldCom an enormous pricing advantage over the competition. They'll be able to cut their interest payments and keep their heads above water. This will also give the current shareholders a major haircut, but who cares about them? We're entering a period where a massive amount of corporate debt will be swapped for equity. For an idea of how this will look, see the bankruptcy of the Penn Central Railroad years ago.

WorldCom's filing will in turn force the telecom suppliers into bankruptcy, as they have extended huge amounts of credit to build out the networks. In bankruptcy, these networks are valued at 3–10 cents on the dollar. Look for Lucent, Cisco, Corning, and similar companies to take heavy hits.

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

Faster information discovery in life sciences

Life-Sciences Standards and the I3C

BY JILL **KAUFMAN, PHD**

Life-sciences standards are just emerging. At standards meetings, competitors, partners, and customers come together to create interoperability interfaces in areas important to customers. IBM is active in life sciences–related standards focused on informatics and data management, many of which use XML and a Web-services approach.

ABOUT THE AUTHOR

Dr. Jill Kaufman is program director of Strategy and Standards at IBM Life Sciences. Jill created IBM's standards strategy and directs IBM's standards work in life sciences. She is an active participant in the I3C. Jill has 20 years of technical and marketing expertise in high tech industries, including telecommunications, computers, and e-commerce.

E-MAIL
jillkauf@us.ibm.com

IBM Life Sciences currently participates in several standards bodies related to life sciences, including I3C (Interoperable Informatics Infrastructure Consortium), OMG-LSR (Object Management Group-Life Sciences Research), HL7 (Health Level Seven), and CDISC (Clinical Data Interchange Standards Consortium). The three focus areas of life-sciences standards are research, development, and clinical genomics. The I3C and OMG-LSR are initially focused on interoperability in the research area, while CDISC and HL7 create standards in development, such as clinical trials. Clinical genomics is an emerging area, and related standards will initially be worked on in HL7. This article highlights the work of the I3C.

I3C

The I3C is the newest life-sciences standards body, incorporated earlier this year. Carol Kovac, general manager of IBM Life Sciences, is on the I3C board along with representatives from Sun Microsystems, Millennium Pharmaceuticals, the Whitehead Institute/MIT Center for Genome Research, and BIO (Biotechnology Industry Organization). The I3C is working to facilitate and enable data exchange and data and knowledge management across the entire life-sciences community. It is developing a common standards-based platform with a use-case approach, based on Web services architecture using technologies such as XML, SOAP, and UDDI. The I3C will produce recommendations for interoperability rather than standards, and will use existing standards (where they exist) as the base.

The I3C will focus initially on genomics. High volumes of data are generated by genomic research from a wide variety of data sources. A common standards-based platform will enable sharing and mining of data, which could speed information discovery in life sciences. The I3C Technical Architecture group is developing protocols, common APIs, and XML-based ontologies that assist with interoperability in life-sciences informatics areas. The use-case approach facilitates putting solutions together faster.

A new I3C work area is the Life Science Identifier (LSID). LSID provides a logical naming convention that defines a means for uniquely naming biologically significant data items, which can help with issues such as inconsistent naming conventions. Most organizations, even at the department level, have their own way of naming individual data items such as sequences, genes, or clones, which impacts integration. Since an LSID is unique for the life of the entity, it makes it easier to track an item through the pipeline.

The LSID provides a common way to identify and access biological data regardless of where it's stored. This location-independent naming scheme separates names from addresses. The LSID enables the creation of scalable systems and facilitates distributed data sharing and search and federated access control.

In the I3C's recent group interoperability demo at the BIO 2002 conference, the de facto standard of Bioinformatic Sequence Markup Language (BSML) was utilized. Created by LabBook. Inc., BSML provides a standard way to encode genomic sequence data and its associated annotations. BSML is XML; it encodes the semantics of the data it represents so the information can be both utilized by humans and processed by machines. Web services uses XML (such as BSML) in the content of the SOAP envelope.

Future I3C work areas that have been discussed include genomics, proteomics, chemical informatics, pathways, clinical genomics, publication data and retrieval, and ontology-driven queries. The I3C has four technical meetings each year, with the remaining meeting in San Diego, November 6–8. Nonmembers are encouraged to attend the first day, which has an education focus. Programming content is often part of these technical meetings. For more information about the I3C, see www.i3c.org.

SITRAKA JCLASS SERVERVIEWS

WWW.SITRAKA.COM/JCLASS/WDJ
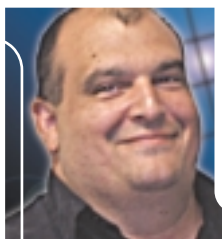
Not just for salads anymore

Scruples

BY JOE **FARSETTA**

Scruples. Business integrity. Honesty. Dealing in a fair and open manner. Trustworthiness. Mutual respect. These words and phrases all describe a mindset that's crucial to the long-term survivability of any services industry. Since IT is primarily a services industry, these words affect the way we do business and the way we're perceived by our clients. It's that simple.

ABOUT THE AUTHOR

Joe Farsetta is an engineer with more than 20 years of industry experience in telecommunications, networking, operations, business process architecture, applications, and support. An entrepreneur and inventor, Joe's past engagements have included Unilever, NJ Transit, and a regional directorship at Bell Atlantic Network Integration. He is currently working as an independent consultant in the New York metropolitan area.

E-MAIL
jjf10965@yahoo.com

After more than 20 years in this business, I've pretty much seen it all. Bone-headed moves run the gamut, from really poor business judgement to half-baked product schemes; from nonexistent planning to price gouging in the name of stupidity. Oh, boy, have I seen it all. Now, not all of it's been bad, and mistakes are made in every business. This is especially true in the IT arena, where explosive growth and technological advances have shaped the industry for the past six or seven years.

With the technological advances came opportunity. And with opportunity came money… lots of money. For those with true business knowledge and savvy, cashing in while providing genuine value and service to the client was a sure bet. For some, though, opportunity was measured in the ability to pump up professional services pricing to new heights. Yes indeed, cash was, and still is, king. The thing that never ceases to amaze me is the way some professional services organizations model their pricing structures and service offerings. While most are reasonable and deliver real value to the client, a few have the apparent inability to distinguish fantasy from reality.

Don't get me wrong. Mitigating risk by adding a few dollars to a quote is perfectly acceptable and, in most cases, necessary. What I'm speaking of here goes beyond the realm of the reasonable. Some providers still believe that customers are dumber than a sack of hammers. They don't merely mitigate risk by adding some dollars. No, no, no… these pretenders price each job as if it were their last!

So, I dedicate this article to those vendors who still don't get it; to those who, after learning they didn't get my business due to excessive pricing, blink their eyes in bewilderment, like a cow staring at a passing train.

This month's article strays from the path of reality and ventures into the realm of silliness. We're traveling to the land of IT Bizarro-World. A land where professional services heroes believe that money grows on trees, and that every client is more clueless than the last. A land where all things are definitely not equal. You know the drill: "No, Mr. Customer, we're not fleecing you. We're adding value…our experts are worth every penny, and we're barely covering our expenses."

Although the scenario I'll depict here is a little far-fetched, it may be closer to reality than you think. So, any similarity to business practices, titles, responsibilities, names, or actual events you may have encountered in your own career are purely coincidental!

In all seriousness, most clients are a whole lot smarter than they were a few years ago. Most aren't willing to be taken to the cleaners in the name of being put "on the Net." The vast majority can smell a bad deal from a mile away. Thank goodness for that – it helps the true professionals separate themselves from the rest. It also gives real professional services organizations the opportunity to outshine the competition. So, relax and read on. Opportunistic or outrageous? You decide!

Once Upon a Time…

Customer X informs Professional Services Company Y that the 100-watt light bulb in their Web-hosting cage is burned out. Customer X can't get his maintenance guy in to replace the bulb until next Thursday, so he asks his account manager to handle the bulb swap. No problem…

Seizing this major opportunity, the account manager quickly engages a business services director from his company, along with a network architect, to review the request. A complete architecture validation is then ordered, as the customer may be confused and not actually need a new bulb. It's decided that the customer needs a truly comprehensive solution rather than a simple bulb replacement, and it's best if he leaves the decision-making to the professionals.

The first thing Professional Services Company Y decides is that, for security reasons, there must be 24 more lighting fixtures installed in the cage immediately. Mind you, this cage is only 10'x10'.

The second thing Professional Services Company *Y* decides is that all new lighting fixtures need to have reversed threads. The sole existing unit requires a retrofit. Additionally, all fixtures need to be fed via dual power sources, for purposes of redundancy. One can never be too careful or too prepared.

A project manager to supervise the fixture installations and retrofit is then added to the project. Six hundred hours are estimated for this critical service. Twenty-five senior consultants are assigned to the project. Their time is estimated at 36 hours per lighting fixture for installation, setup, documentation, testing, and certification. Remember, these are highly trained individuals, prepared for any emergency that may arise during the critical "installation phase."

As the completion of this critical reengineering exercise approaches, a final architecture review and approval

the required candlepower to outshine the sun. Scientists from NASA's Jet Propulsion Laboratory are retained to validate the new design. Professional Services Company *Y* has created yet a new offering for its customers… they call it *Jumbo Jet*, referring to the price tag for the service, and to the labs where the validation is to be reviewed. Pure genius, don't you think?

Solar deflectors and ultraviolet diffusers are then special-ordered to complement each laser and evenly distribute the light within the cage.

With safety as a concern, specially designed UV-, EMI-, and photon-protective entry suits are ordered for the senior consultants who will work in the cage, lest they be instantaneously burned to a crisp or perish from radiation exposure.

Finally, water-cooled and lead-lined titanium panels are installed all around the cage to protect anyone in the vicinity from the tremendous

composure, he threw his account manager out the window.
(d) All of the above

If you answered "d," you were correct…

Is there a lesson to be learned here? I think so. Did this scenario actually happen? Of course not… but it might as well have. In my long IT career, I've seen the most outrageously inflated pricing models forced onto customers on a regular basis. In some companies, this was the norm rather than the exception. And what did this over-inflation ultimately do to those businesses? It ruined them, plain and simple. Once you break the customer's trust, it can never be restored. Without trust, there is no additional business. Without additional business, you can turn off the lights and shut the doors.

We're on the cusp of a new breed of applications and client business models. We, the industry professionals, are

Once the battle lines have been drawn, and companies are engaged to perform complex WebSphere implementations, the truth shall set the customers free

must be obtained from Professional Services Company *Y's* chief Internetworking designer in charge of technologies (IDIOT, for short) before presenting the designs to the customer for approval and proceeding with the project. But a snag has occurred!

Upon review, the chief IDIOT sees major problems with the plan. It must be scrapped immediately! The first red flag noted is that the project has no code name – and code-named projects are worth more, right? Of course they are! The project is then appropriately named *Project Catapult*. Pretty snazzy, wouldn't you say?

The chief IDIOT then decides to replace the 25 porcelain light fixtures originally specified with 14 high-end network routers. He'll fully populate each box with single-mode optical fiber interfaces. His thought isn't to use the routers for any network function per se, but to utilize the lasers in each fiber-optic interface to produce

heat, radiation, and blinding light of the artificial sun that's been created.

Four thousand more man hours are added to the cost of the project.

A 93-page statement of work and comprehensive project plan is created and delivered to the customer. The total project cost is estimated at $15.3 million dollars…Project Catapult has successfully catapulted the customer into the arms of another professional services organization. The account manager is left with a huge "L" tattoo on his forehead, and so it goes.

Pop Quiz
In this story…
(a) Professional Services Company *Y* deserves to be run out of town on a rail.
(b) Professional Services Company *Y* believes that Customer *X* was born last night.
(c) After reviewing the quote, Customer *X* laughed until he fell down. Then, after regaining his

poised to help these clients and truly make a difference. It's both an awesome responsibility and an unbelievable opportunity to make some serious dough. So, don't sell yourself short, but don't gouge the customer either. Most important, deliver a quality product on time and within budget. This is true value-add, and is the key to retention, positive reputation, repeat business, and financial success. Once the battle lines have been drawn, and companies are engaged to perform complex WebSphere implementations, the truth shall set the customers free. Pretenders will quickly fail. Experts delivering value-add will prosper. When the smoke clears, you want to be one of the last ones standing. Integrity and knowledge are your most important tools. Leave amateur hour to the pretenders and hacks. They will certainly fail.

Above all, remember that scruples are not things that you put on your salad. ⊕

KINETIKS
WWW.KINETIKS.COM

IBM Life Sciences

Framework

WRITTEN BY
MICHAEL **NIEMI**

Application and data integration in the life sciences

ABOUT THE AUTHOR

Mike Niemi is a senior software engineer in the IBM Life Sciences organization. He has held various positions in software and hardware development over the past 24 years. His experience includes development in voice systems, WebSphere Application Server and SiteAnalyzer, TCP/IP, and mainframes.

E-MAIL

niemi@us.ibm.com

Pharmaceutical companies are facing the challenge of improving the productivity of the drug discovery and clinical trials process, creating and sharing knowledge across the silos of that process, and integrating applications and data in enterprise-wide development efforts. Biotech, research, and medical organizations face similar challenges of collaboration and sharing of applications and data.

To help address these challenges, the IBM Life Sciences Framework uses industry-standard technologies (J2EE, XML, Web services, etc.) and protocols and data representations from standards efforts such as the I3C (Interoperable Informatics Infrastructure Consortium), OMG-LSR (Object Management Group-Life Sciences Research), HL7 (Health Level 7), and the Bio* projects. The framework addresses the integration of applications, data, and user interfaces.

The Convergence of Life Sciences and Information Technology

The application of information technology to the life sciences is essential to the progress of medical research, the drug discovery process, and the realization of improved health care. Awesome innovation is occurring in

this area. In "Creating a Bioinformatics Nation," Lincoln Stein compares the old city-states of Italy to bioinformatics today. Determining how to assist in the productive and timely use of this innovation, so it proceeds in a parallel process, is a fundamental challenge. The results of the individual "city-states" (individuals, departments, companies, products) can be amplified though the use of common interfaces. This is a problem of coarse-grained, loosely coupled integration (from the IT perspective) of applications and data. The use of broad industry and community standards will enable this.

This article provides a simple example of accessing a Web service called XEMBL. XEMBL is a means of accessing EMBL nucleotide sequence data. This is a publicly available database kept at the European Bioinformatics Institute (EBI). In a future article we'll look at using a UDDI registry to allow service providers and requesters to share services in this sample application.

XML Vocabularies in the Life Sciences

XML as a standard data exchange format provides a great deal of power to support program-to-program inter-

action. It's also the obvious choice for several reasons. First, XML allows for the creation of tags to define the semantics of a particular XML vocabulary, so we can build vocabularies intended to be useful in specific domains such as math, chemistry, and genomics.

Second, the creation of tag sets in XML is relatively simple and fast. Because the structure of XML is standard, the data is self-describing and can be interpreted and parsed by machines.

Third, tags delimit the content and syntax to allow us to build data structures of arbitrary size and complexity, and the data is delivered in text format. Text is very good for exchanging information across diverse platforms, since virtually every system can handle it.

Finally, industry support – all the major vendors are accepting XML as a standard for exchanging data. Many vocabularies are emerging across the drug discovery domain. Infrastructure support is available to translate among the various vocabularies. Ontologies are being developed to allow for machine-to-machine interaction in this space.

Figure 1 shows some of the XML vocabularies in the life sciences.

XML, Web Services, Etc.

Web services is another standard that has broad industry support. Some fun sample Web services can be found at XMethods (www.xmethods.org). WebSphere has strong support for Web services in WebSphere Application Server (WAS), WebSphere Studio Application Developer (WSAD), and across the product line.

Figure 2 depicts something that's probably pretty familiar to you – an *n*-tier architecture. Clients communicate using Web browsers or Internet-enabled applications with an application server. WAS is an example of a Tier-1/Tier-2 middleware layer. The Web services support built into WAS and tools such as WSAD help with application integration.

The applications running in the middle tier access databases using SQL queries via JDBC. IBM DiscoveryLink is a convenient way to access distributed heterogeneous data sources. It uses sophisticated query optimization to access those data sources and helps with data integration.

Large multinational pharmaceutical companies want to provide a consistent look and feel for their enterprise-wide applications. Mergers, widely separated development groups, and outsourcing of development projects tend to make this more difficult. A portal, such as WebSphere Portal, helps with integrating the user interface.

One of the organizations working on standards for interoperability in the life sciences is the I3C. IBM is one of its founding members and helped to develop interoperability demonstrations for the 2001 and 2002 BIO conferences. Figure 3 shows the configuration developed for BIO 2002.

The figure shows the client applications interoperating with one another using a common XML vocabulary, called BSML, to represent gene sequences. SOAP is used to invoke Web services on the application server. The Web services use JDBC and SQL to make queries on the data sources. IBM DiscoveryLink allows the application to present a complex query to a federation of distributed heterogeneous databases.

A Simple Example of a Web Service in the Life Sciences

Here you'll see a simple example of invoking a Web service, XEMBL, in the life sciences. XEMBL and the Open Bibliographic Query Service, both hosted at the EBI, are examples of the loosely coupled, services-based (r)evolution that is occurring.

The XEMBL Web service takes two parameters – the accession number for the sequence of interest and the XML format that you want returned. The accession number is a unique identifier for a sequence record. Currently XEMBL supports BSML and AGAVE XML formats for the result. In this example we'll ask the service for the BSML format.

Listing 1 shows the code for a Java client that invokes the XEMBL Web service. We used IBM's Java IDE, WSAD, to develop this sample (see Figure 4). WSAD makes it easy to develop client-side code as well as the full suite of Web services, including services to run on WAS, the WSDL to describe those services, and the use of UDDI to publish and discover those services.

It's possible to develop an application by browsing a UDDI registry and importing services from it. Import the WSDL into your project, then create a skeleton JavaBean and generate a Java client proxy and a sample application from the WSDL document. You can then easily test the code using the integrated debugger.
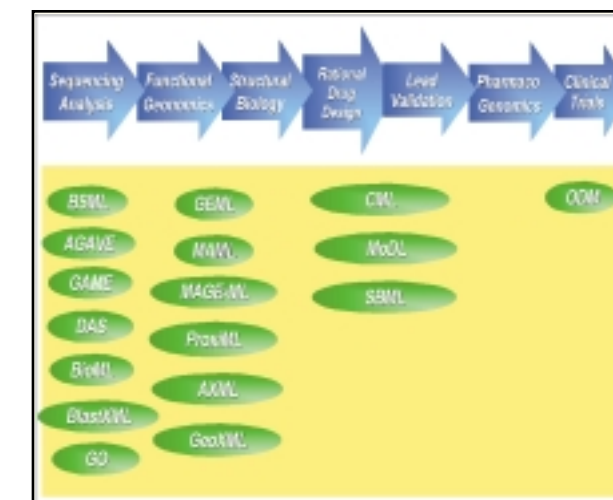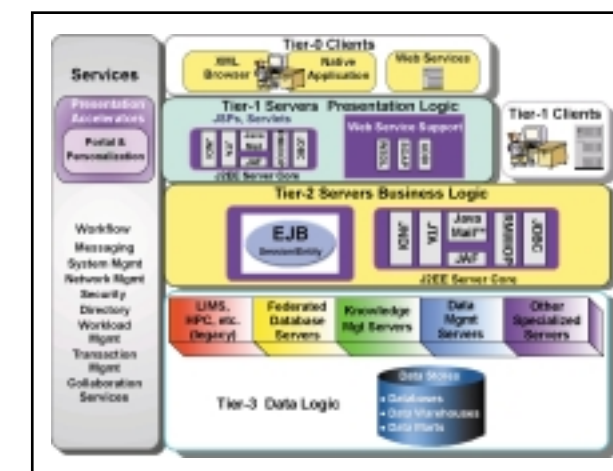

FIG. 1: XML VOCABULARIES

FIG. 2: N-TIER ARCHITECTURE

Listing 2 shows a portion of the response. It's an XML document containing the nucleotide sequence we requested. The "agct"s that you see in the <Seq-data> element are the alphabet soup of life. Also in the document are the history and annotations on that sequence. Annotations mark locations of biologically important parts of the sequence.
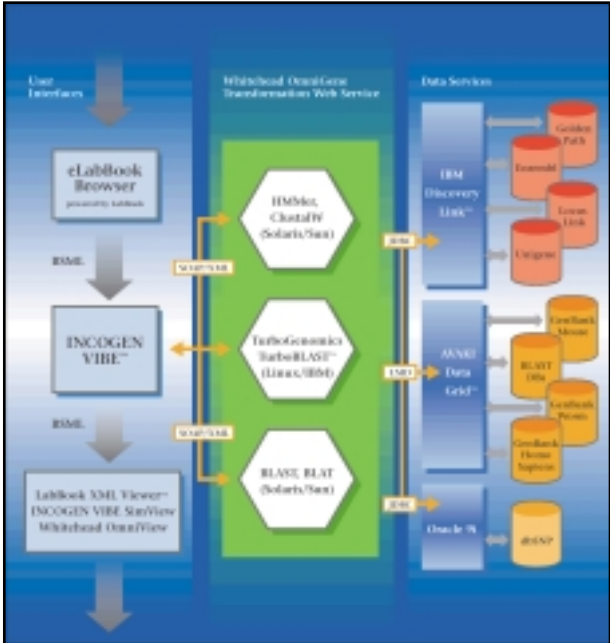

FIG. 3: CONFIGURATION DEVELOPED FOR BIO 2002

Listing 3 shows a simple Perl script that uses the SOAP toolkit SOAP::Lite for Perl. It uses the WSDL file from the EBI site to create the service. Perl IDEs help users include Web services in their Perl code. In these IDEs, a Web services popup wizard can be presented to the user. The user can point to the WSDL for the Web service he or she is interested in (in this case the WSDL is on the EBI site), and the Perl IDE can then assist in selecting the method for that Web service and setting the parameters to it. The script can then be tested quickly in the IDE using the Web service.

Summary

In this article we touched on some of the challenges facing the life sciences industry. For more on this see "The Life Sciences Revolution" article elsewhere this issue. We've discussed the need for standards in addressing these challenges; this includes technology standards (J2EE, XML, Web services, etc.) as well as domain standards (coming from the I3C, OMG-LSR, HL7, the Bio* projects, etc.). Finally, we looked at some code to access a Web service at EBI.

References

- Stein, L. (2002). "Creating a bioinformatics nation." *Nature*. May. www.nature.com/cgi-taf/DynaPage.taf?file=/nature/journal/v417/n6885/full/417119a_fs.html&content_filetype=PDF
- XEMBL: www.ebi.ac.uk/xembl
- Open Bibliographic Query Service: http://industry.ebi.ac.uk/openBQS
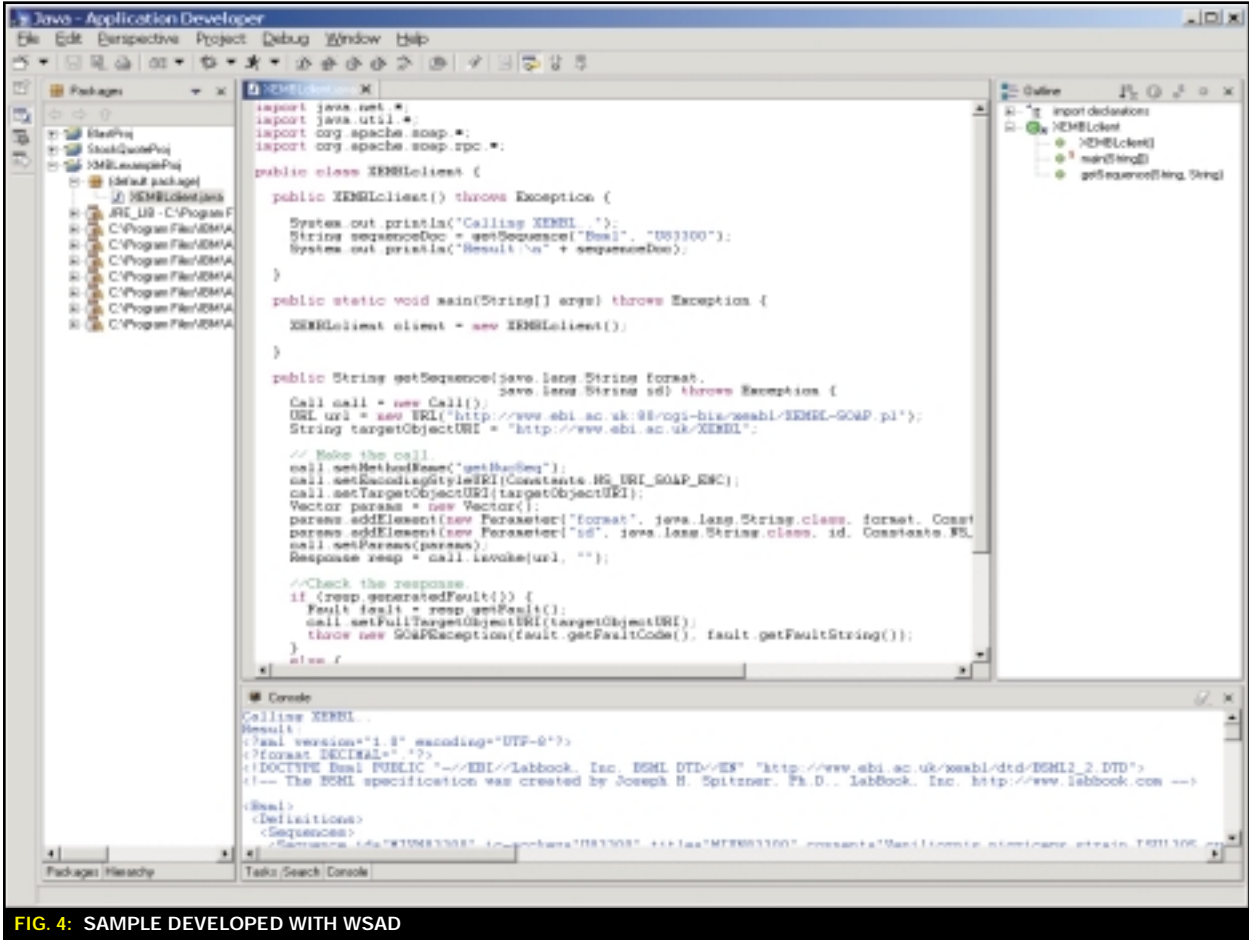- SOAP::Lite for Perl: www.soaplite.com

FIG. 4: SAMPLE DEVELOPED WITH WSAD

CANDLE

WWW.CANDLE.COM/WW1/WEBSPHERE_TRIAL

- developerWorks: www.ibm.com/developerworks/webservices
- I3C: www.i3c.org
- Pacholski, P. (2002). "Web Services Programming with WebSphere Studio Application Developer" April. www7b.software.ibm.com/wsdd/library/techarticles/0204_pacholski/pacholski.html
- XMethods: www.xmethods.org

IBM Products and Solutions

- WebSphere Application Server: www.ibm.com/software/webservers/appserv
- WebSphere Studio Application Developer: www-3.ibm.com/software/ad/studioappdev

- WebSphere Portal: www-3.ibm.com/software/web servers/portal
- IBM Life Sciences Framework: www.ibm.com/solutions/lifesciences/framework/index.html
- IBM DiscoveryLink: www-3.ibm.com/solutions/lsframework

For More Information

- IBM Life Sciences: www.ibm.com/solutions/lifesciences
- IBM Life Sciences Framework Presentation: www.ibm.com/solutions/lifesciences/pdf/FW_Sales_pres entation_V2.pdf
- Informatics: A Key to Success in the Life Sciences Industry: www.ibm.com/solutions/lifesciences/pdf/Informatics.PDF 🌐

LISTING 1

```
import java.net.*;
import java.util.*;
import org.apache.soap.*;
import org.apache.soap.rpc.*;

public class XEMBLclient {

  public XEMBLclient() throws Exception {

    System.out.println("Calling XEMBL..");
    String sequenceDoc = getSequence("Bsml", "U83300");
    System.out.println("Result:\n" + sequenceDoc);

  }

  public static void main(String[] args) throws Exception{

    XEMBLclient client = new XEMBLclient();

  }

  public String getSequence(java.lang.String format,
                            java.lang.String id)
                                        throws Exception
{
    Call call = new Call();
    URL url = new URL
("http://www.ebi.ac.uk:80/cgi-bin/xembl/XEMBL-SOAP.pl");
    String targetObjectURI = "http://www.ebi.ac.uk/XEMBL";

    // Make the call.
    call.setMethodName("getNucSeq");
    call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
    call.setTargetObjectURI(targetObjectURI);
    Vector params = new Vector();
    params.addElement(new
            Parameter("format", java.lang.String.class,
                      format, Constants.NS_URI_SOAP_ENC));
    params.addElement(new
            Parameter("id", java.lang.String.class,
                      id, Constants.NS_URI_SOAP_ENC));
    call.setParams(params);
    Response resp = call.invoke(url, "");

    //Check the response.
    if (resp.generatedFault()) {
      Fault fault = resp.getFault();
      call.setFullTargetObjectURI(targetObjectURI);
      throw new SOAPException(fault.getFaultCode(),
                              fault.getFaultString());
    }
    else {
      Parameter refValue = resp.getReturnValue();
      return ((java.lang.String)refValue.getValue());
    }
  }

}
```

LISTING 2

```
<?xml version="1.0" encoding="UTF-8"?>
<?format DECIMAL="."?>
<!DOCTYPE Bsml PUBLIC "-//EBI//Labbook, Inc. BSML DTD//EN"
 "http://www.ebi.ac.uk/xembl/dtd/BSML2_2.DTD">
<!-- The BSML specification was created by Joseph H.
 Spitzner, Ph.D., LabBook, Inc. http://www.labbook.com -->

<Bsml>
     <Definitions>
   <Sequences>
    <Sequence id="MIVN83300" ic-acckey="U83300"
     title="MIVN83300" comment="Veniliornis nigriceps
     strain LSU1305 cytochrome b gene, mitochondrial gene
     encoding mitochondrial protein, partial cds."
     length="946" topology="linear" molecule="dna">
                            ...
     <Seq-data>
        agcttctgggcttccagacccagctactttgcggaactcagcaacccagg
        catctctgagtctccgcccaagaccgggatgcccccaggaggtgtccgg
        gagcccagcccttcccagatagcagctccgccagtcccaagggtgcgcaa
                            ...
        tgggggctggcctctggctctcatggggtccaactttttgtgtattcttca
        acctcattgacaagaactgaaaccaccaatatgactcttggcttttctgt
        tttctgggaacctccaaatcccctggctctgtcccactcctggcagca
     </Seq-data>
    </Sequence>
   </Sequences>
     </Definitions>
</Bsml>
```

LISTING 3

```
#!/sw/arch/bin/perl

use SOAP::Lite;

# Create the web service from the WSDL
my $service = SOAP::Lite
       -> service('http://www.ebi.ac.uk/xembl/XEMBL.wsdl');

# Call the service
my $result = $service->getNucSeq("Bsml", "U83300");

# Print result
if ($result) {
  print "Result:\n" . $result;
} else { print "Error: " .
             "faultcode=" . $service->call->faultcode .
             ", faultstring=" . $service->call->faultstring;
}
```

ALTAWORKS

WWW.ALTAWORKS.COM

A simple alternative

Building DB2-Based Web Services Using WebSphere PART 1

BY RON **BEN-NATAN**

I've been involved with Web services for more than a year now. After the initial fascination and learning curve that are part of any new technology comes the part where you roll up your sleeves and start applying it for the sake of solving real problems or making architectural improvements to an existing system – as opposed to applying the technology for the sake of applying the technology.

ABOUT THE AUTHOR

Ron Ben-Natan, CTO of ViryaNet Inc., holds a PhD in computer science in the field of distributed computing and has been architecting and developing distributed applications for more than 15 years. Ron's hobby is writing about how technology is used to solve real problems. He has authored numerous books, including *IBM WebSphere Application Server: The Complete Reference*, published by Osborne/McGraw-Hill.

E-MAIL

rbennata@hotmail.com

Idiscovered early on that many real Web services applications involve database access. In many cases, Web services serve as an access layer, allowing functionality implemented in database management systems (whether through the use of SQL commands or through stored procedures) to be exposed and deployed for Web access. In such cases Web services act as wrappers to existing database-centric systems – wrappers that may be accessed over a SOAP or HTTP connection and that use XML to package inbound and outbound data. The pattern of Web services wrapping database operations stems from the fact that Web services make excellent wrappers and that many of the systems out there are database-centric.

Given that Web services are often used as wrappers for existing functionality, the main question is what should you be wrapping as a Web service. Normally the answer is a set of methods or functions deployed in the application server. My article

"Developing Web Services with WebSphere Studio" (*WSDJ*, Vol.1, issue 4) showed how to use WebSphere Studio Application Developer to wrap an existing Java method as a Web service without writing a single line of code.

This month I'll introduce a technology that answers the question differently – one that allows you to quickly and easily wrap functionality within a DB2 database as a Web service. Because many production systems have evolved (or are in the process of evolving) from a client/server architecture to some form of three-tier architecture, the database server often encapsulates many of the business functions implemented in the system. In these cases, directly wrapping the database with Web services may be a simpler alternative that doesn't require you to first wrap database operations within Java methods and then create the Web services wrappers – instead, everything is done for you based on database operations and metadata.

The reasons for wrapping these APIs as Web services can be summarized as follows:
1. The APIs need to be accessed by multiple decoupled systems, each created by a different vendor.
2. Remote invocations may occur over the LAN or WAN. WAN access may

Usage Scenario

Before delving into the technical details, let's look at a usage scenario taken from the world of call centers. Imagine a call center database in which all problem reports and work statuses are stored. For every problem called in by a customer, a CALL is created. The information describing the call is maintained in the CALL table. Once the call is logged, the support center starts working on it. Work is encapsulated in an ACTION, which records information such as who's assigned to handle it, the scheduled start date, the estimated time of arrival (ETA), and so on. Information pertaining to what work was done in the context of an action is also recorded, and all this is maintained as a record in an ACTION table.

The CALL and ACTION tables are the heart of the service operation. Customer information systems need to access this data in case a customer calls and wants to know the status of an open problem; field engineers need to access it to get their work list for the day; managers need to run reports to track the number of open calls and how long it took to resolve problems; and financial systems need to access the action details to figure out whether the customer needs to be billed for the work. Some of these systems may even be located outside the enterprise. For example, it's possible that the field engineers work for an external service provider and use a separate system for work management. In this case, access to the central database is complicated by organizational boundaries, firewalls, permissions, and so on. It therefore makes sense to provide the API layer as a set of Web services.

The reasons for wrapping these APIs as Web services can be summarized as follows:
1. The APIs need to be accessed by multiple decoupled systems, each created by a different vendor.
2. Remote invocations may occur over the LAN or WAN. WAN access may

include a third-party contractor managing an independent workforce, as well as wireless access from field units used when field engineers access their work list for the day. Figure 1 shows receipt of work orders though a Web-service interface delivered over a wireless network. The work orders communicated to the field include details on the scheduled start date, the work to be completed, and more. All this data is available within the CALL and ACTION tables, for example, the work order shown in Figure 1 (3975/1) is ACTION 1 in CALL 3975.
3. Access to this data needs to come from multiple platforms. Each system may be written using a different programming language and running over a separate operating system. Because all modern platforms have added (or are adding) support for Web services, this makes it the only true interoperability technology today. This includes all platforms.

As an example, Microsoft recently released a new toolkit for Office XP that allows developers to consume Web services. It contains the Web reference tool for Visual Basic for Applications, which can be used to create proxy classes based on WSDL files. If you want to use Excel to display all call details for a certain customer, you can create a Visual Basic (VB) Web service proxy that invokes the APIs deployed on WebSphere Application Server (WAS) to access data stored in DB2 – all without writing a single line of code. Figure 2 shows an Excel spreadsheet with a push button used to call the VB stub that invokes the Web service and brings back the call information based on a customer name.

DB2 Web Services Wrappers

To make the discussion concrete, let's look at a small set of APIs to be delivered as Web services. (These Web services will be defined and implemented in my next article.) The operations to be wrapped as Web services are:
- Given an employee ID, return a list of all work assigned to this resource.

FIG. 1: DETAILS DELIVERED OVER A WIRELESS NETWORK TO A MOBILE DEVICE

- Given a call number and an action number, return the status of the action and the status of the call (these may be different because a call may have multiple actions).
- Update the status of an action.
- Retrieve all call details.
- Given an employee ID, call a stored procedure that computes the yearly first-time fix rate (i.e, how frequent-

ly the engineer resolves the problem on his or her first attempt). This complex business function cannot be phrased as an SQL operation and is implemented as a stored procedure.

These functions are based on two tables: the CALL table and the ACTION table. Some of the column names and descriptions are shown in Table 1.

Given these (partial) table structures, it's easy to implement the required functionality using simple SQL statements. As an example, the following SQL command returns a list of call/action numbers assigned to an employee with an ID of 112001:

```
select call__t, action_no
from action where
action_status__c!='CLOSE'
    and
who_action__c='112001'
```

This returns a collection of 2-tuples, each with a call number and an action number:

```
3922  1
4006  1
3974  1
4037  1
3529  1
512   1
3975  1
```

This addresses the required functionality, but it's not exactly a Web service, is it? Here's where the magic
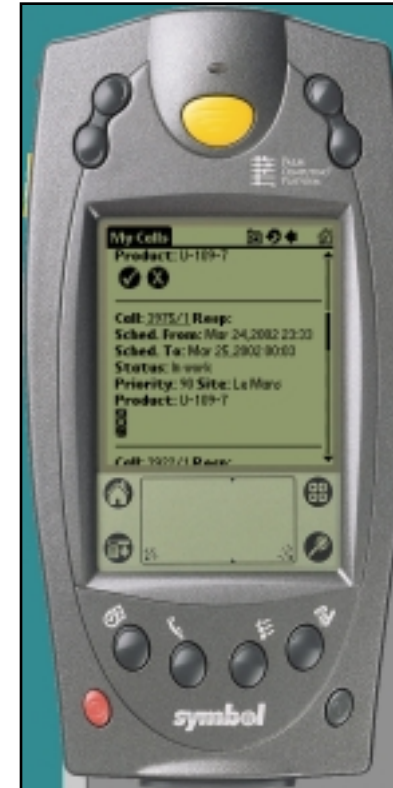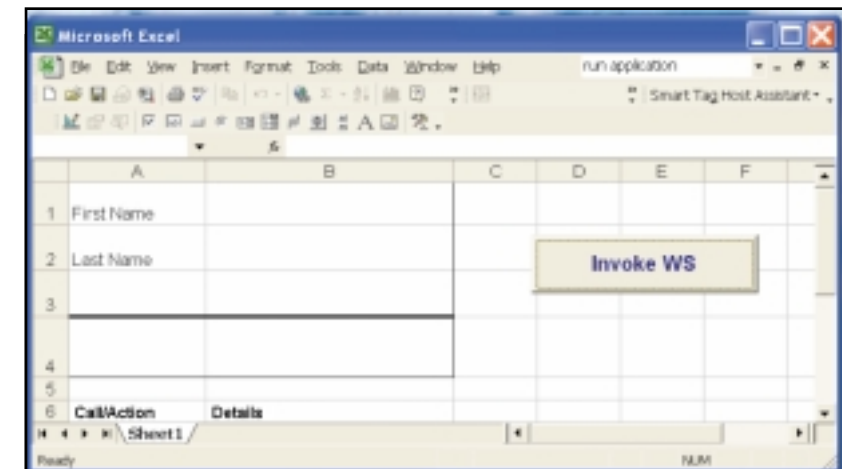

FIG. 2: INVOKING THE WEB SERVICE FROM AN OFFICE APPLICATION

CALL	ACTION
CALL_T CALL NUMBER	**CALL_T** CALL NUMBER
BOOK_DATE DATE ON WHICH THE CALL WAS BOOKED	**ACTION_ NO** ACTION NUMBER WITHIN THE CALL
SYMPTOM_C SYMPTOM CODE DESCRIBING THE PROBLEM	**WHO_ACTION_C** EMPLOYEE ID OF THE RESOURCE ASSIGNED TO HANDLE THIS ACTION
PRIORITY CALL PRIORITY	**ETA_DATE** ESTIMATED TIME OF ARRIVAL
ACT_RESP_TIME ACTUAL TIME THAT THE COMPANY RESPONDED TO THE CALL	**ACTION_STATUS_C** STATUS CODE FOR THE ACTION
PROBLEM_DESCRIPTION	
CALL_STATUS_C STATUS CODE FOR THE CALL	

TABLE 1: COLUMNS IN THE CALL AND ACTION DB2 TABLES

starts. Using DB2, WAS, and a technology package called Web Services Object Runtime Framework (WORF), this can be immediately packaged as a Web service wrapper to be deployed on WAS accessing DB2.

WORF is a set of tools for implementing Web services with DB2. It is deployed on WAS and uses Apache SOAP 2.2. It implements a layer that runs on WAS and is responsible for taking database access definitions and translating them (on-the-fly) to Web services constructs, including SOAP messages and WSDL documents. The mapping between the database definitions and the Web service is done in a Document Access Definition eXtension (DADX) file. WORF uses the DADX definition to provide an implementation of a Web service through a servlet that accepts a Web service invocation over SOAP, an HTTP GET, or an HTTP POST. This servlet implements the Web service by accessing DB2, invoking the SQL operation defined in the DADX file and returning the results as a SOAP response (see Figure 3).

The DADX file used to build a Web service for the above query is shown in Listing 1. The file encapsulates the SQL used to implement the Web service functionality and allows for the employee ID to be passed in as an argument. To make this Web service available on your WebSphere server (assuming you have WORF and Apache SOAP 2.2), simply place this file under a subdirectory within the WEB-INF tree in the WORF-installed application directory (more on this in Part 2). That's it – your DB2 SQL command is now a Web service! You can invoke it using a SOAP request delivered to a URL of the form:

```
http://<host>/services/sam-
ples/getWorkForEmployee.dadx
/SOAP
```

You can also invoke the service using an HTTP request of the form:

```
http://<host>/services/sam-
ples/getWorkForEmployee.dadx
/ getWorkFor
Employee?employee_id=112001
```

Summary

WORF allows you to take the investment you've made in your DB2 schema and procedures and immediately leverage it using Web services deployed on WAS. This first article of a two-part series explained the general framework of WORF and DADX and introduced you to a business scenario that illustrates this technology. Part 2 will continue with this scenario, including the technical details on how to build your DADX files, how to deploy them on WAS, and what other configuration settings are necessary to make your DB2 investment pay off as Web services. 🌐

FIG. 3: IMPLEMENTING A WEB SERVICE USING WORF, DB2, AND WAS

LISTING 1
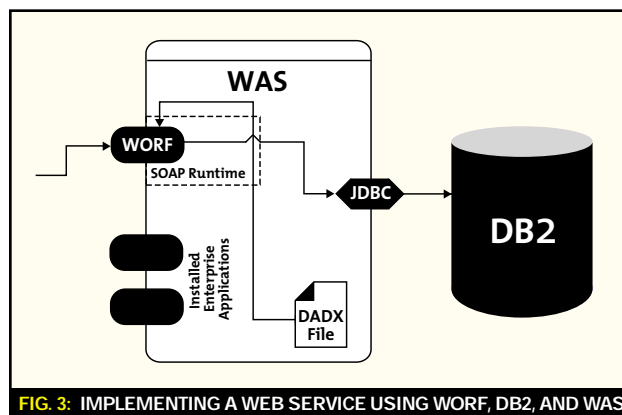
```
<?xml version="1.0" encoding="UTF-8"?>
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx">
 <operation name="getWorkForEmployee">
  <query>
   <SQL_query>
    select call__t, action_no from action
    where action_status__c != 'A501'
    and who_action__c = :employee_id
   </SQL_query>
   <parameter name="employee_id" type="xsd:string"/>
  </query>
 </operation>
</DADX>
```
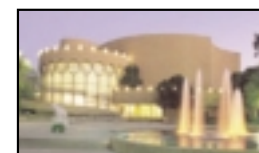

Talking WITH Sharon Nunes

Jack Martin, editor-in-chief of WSDJ, recently spoke with Dr. Sharon Nunes, IBM's director of Life Sciences Solution development, about how pharmaceutical companies are handling the rapid growth of information in drug development.

WSDJ: HOW DID IBM GET INVOLVED IN THE PHARMACEUTICAL INDUSTRIES AND WHAT TYPE OF PROBLEM WERE YOU TRYING TO SOLVE?
Sharon: In the mid '90s, IBM Research was collaborating on a few joint projects with pharmaceutical companies in the area of drug design and in the area of creating algorithms for pattern discovery and pattern matching. These algorithms helped researchers find patterns in the gene sequences that were coming out of the Human Genome project. There wasn't a huge amount of data at that time compared to what we have today, but we realized that they were generating huge amounts of data. IBM has always been engaged with pharmaceutical companies on the business side, but this was a new, exciting opportunity for us – IBM Research could create new tools to help analyze data being generated around the Human Genome project.

In the late '90s IBM research formally launched its Computational Biology Center, which now has about 50 people around the world doing various aspects of research related to the computational challenges of pharmaceutical companies. Today we also have a large business organization with almost 200 people worldwide in the core group, as well as another 500 in the IBM brand groups, who are dedicated to life sciences activity. I manage the worldwide solutions development activity, which has an extended team of about 100 people. Our focus in Life Sciences is to deliver solutions to the pharmaceutical and biotech industries. Our solutions target high-performance infrastructure, data and knowledge management, and clinical development/regulatory compliance.

WSDJ: WHAT ARE SOME OF THE KEY ISSUES THAT YOU'RE TRYING TO SOLVE FOR PHARMACEUTICALS, THINGS THEY ARE FACING TODAY?
SN: Pharmaceutical companies are faced with an explosion of information, more so within the last five to seven years. Let me give you some specific data points. There is a technology called microarray analysis; you can get 40,000 data points off one of these microarrays. Think of a silicon chip because the technology is similar, but this has DNA strands on it; you can look at the interaction of DNA strands with a possible drug for example.

There are also many public databases that hold gene sequences and protein sequences and protein structures. One of the more popular ones, GenBeg, is growing at the rate of almost a million entries an hour. In addition, pharmaceutical companies and biotech companies themselves are generating millions of data points in lab experiments every week, whether it's mass spec experiments that they are running in the lab to do some protein analysis or they are doing some gene sequences themselves. You need sophisticated computer programs to look at the structures of these proteins and the possible interactions of the proteins with other proteins in the body or with proteins and drugs. You need huge amounts of computer storage to hold all that information, and you need smart analytical programs to make sense out of that data and to allow you to access the right data for your business decision. So, there's a real explosion of information around the Genome Project and the work that's come along subsequently relating to proteins and protein structures.

Proteins are really important because it's the interaction of drugs with the proteins in the body that helps to either minimize the effects of diseases or to cure them. Typically, when the protein malfunctions, you see the onset of disease. It's important for researchers to understand the effects of proteins in the body, and the pharmaceutical company will develop drugs targeted at a specific protein to react in a certain way to either cure a disease or to minimize its effects. The explosion of information is one of the challenges as companies grow, and there has been a tremendous number of mergers and acquisitions in the last several years with the pharmaceutical companies.

Management of information, sharing of information, collaboration, and knowledge management are all key issues that are challenging our customers. There are many "point applications," and we are hearing that integration of data and integration of applications is another challenge for our customers. The IBM Life Sciences Framework is IBM's solution to this problem that is so critical to our customers.

WSDJ: WHAT ARE THE TWO BIGGEST THINGS KEEPING PHARMACEUTICAL COMPANIES FROM BEING SUCCESSFUL TODAY?
SN: One of them, I really think, goes back to what I said regarding the sharing of information and collaboration. One of our customers said, "I have nine labs around the world. I know I have researchers in all of those labs who are probably working on similar projects. How do I get those researchers to find each other and to communicate with each other and to share information?" We believe that with IBM's Life Sciences Framework, the WebSphere Portal, for example, can make it easier for them to share information. We have what we think are cool tools to enable finding the right resources and organizations automatically that really can help the customers solve some of those problems.

Another key challenge relates to data integration and application integration because each customer we deal with has a number of favorite applications. There are thousands of applications in the biology and chemistry domains that our pharmaceutical customers like to use. One of the challenges is how to get the chemistry data to talk to the biology data as you proceed down the drug development process. How do you reach back and get information from the beginning of the pipeline that may be critical for you to make a good decision in toxicology, for example, testing the toxicity of a drug in either animal or human trials.

One of the things we have done in our framework is focus on open standards. This is very much in line with the IBM e-business applications framework. We call it the Life Sciences Framework and it focuses on open standards, and an open, modular architecture. WebSphere and WebSphere Portal are both key components of it. We're looking at integration of these multiple applications through XML, which is becoming very much a standard in the life sciences space. We think that these technologies are key for enabling our customers to follow the flow of information through their enterprise pipeline. As you move from biology to chemistry to clinical domains during drug development, you want to be sure you can readily retrieve data from any part of those processes. We believe it is critical to have open standards and Web services, which will allow any application to talk to another application.

WSDJ: DOES THIS ACCELERATE THEIR TIME TO DISCOVERY?
SN: We believe it can make a tremendous difference. We've used the e-business applications framework in many other domains and have information from other areas where it has enhanced productivity. We have developed some prototypes and we are actually in pilots with three customers right now using various aspects of the IBM Life Sciences Framework. We've simplified a lot of the operations into a kind of automated back-end script that enables you to automate a lot of the manual processes. Now researchers might say, "You know I can't have everything automated because there are certain points where I need to add some of my own domain expertise." IBM has a huge number of domain experts and chemists/biologists with industry experience. We have consulted with them during the development of the Life Sciences Framework and, with their guidance, have developed a prototype that automates as much as possible, but also presents options for decision-making to the scientists at certain points along the development process. We have a combination of human intervention with background automation, which simplifies a lot of the manual processes that have been used in the past.

WSDJ: DO YOU HAVE ANY SENSE OF HOW MUCH TIME IT SAVES THEM FROM A PERCENTAGE STANDPOINT?
SN: We have just implemented some early prototypes. It's hard to make a concrete prediction, but one of our customers has implemented a piece of this framework in their business environment. We've heard from them that a process which formerly took them about a week now takes them just a couple of hours. To me, that's a significant productivity increase if you can cut off four or five days from part of your development cycle. That's a tremendous savings in time of people and time to market as you start to march down the development path.

WSDJ: WHY IS THIS APPROACH BETTER THAN THE COMPETITION'S AND HOW IS IT UNIQUE?
SN: We believe that IBM has a true end-to-end, enterprise-based framework. We also think that we bring in the best applications through an open framework so our customers can decide what applications they want to use as their

front end to integrate on top of our framework. We do not require that all of the applications run just on an IBM platform because if you can integrate them with XML or Web services, they can run on competing platforms. We don't like to encourage that, but of course, that's the reality of most of this industry.

We think that we offer more flexibility to our customers by offering this open environment and we also believe that through our data integration software called DiscoveryLink – which is part of our framework – that we enable our customers to integrate heterogeneous data as well. The data does not all have to be in a relational format. It can be in a spreadsheet on someone's desktop, it can be in a flat file, it can be data that's accessed over the Web, or it can be in a relational format. It can even be in a competing relational database format, not just in our DB2 format. We believe that those two factors – access to heterogeneous information and the flexibility of integrating any application into our framework – really differentiate us from any of the competing systems that exist today.

WSDJ: ARE YOU THE ONLY ONE ON THE PLAYING FIELD THAT'S BASED ON OPEN SYSTEMS RIGHT NOW? IS THAT WHAT MAKES THIS SO SPECIAL COMPARED TO ANYTHING THAT A PHARMACEUTICAL COMPANY CAN DO ELSEWHERE?
SN: Well, the IBM Life Sciences Framework is open and modular. We believe there's a tremendous amount of flexibility. We don't run on a closed proprietary system; we think that we offer the best openness and flexibility to our customers compared to any other systems out there today. In addition, our IBM Life Sciences Framework really is

an embodiment of the I3C standards, which are focused on the interoperability of systems and infrastructures based on the needs of the life sciences and pharmaceutical industries. We believe that because of this openness, flexibility, and focus on standards, IBM has an incredibly powerful solution for our customers.

I think one of the reasons why something like this framework and enterprise-based system is important is when we talk to our customers about their own costs, it takes 10 to 12 years to develop a drug, including the clinical trials and approval through the FDA. Every day that they can get a drug to market sooner means a million dollars of opportunity in their pockets. There is a tremendous financial incentive to accelerate the drug discovery process as much as possible.

WSDJ: WHAT WOULD A PHARMACEUTICAL COMPANY FIND OVER THE WHOLE 12-YEAR PERIOD IT TAKES TO GO FROM CONCEPT TO DRUG APPROVAL BY THE FOOD AND DRUG ADMINISTRATION?
SN: The numbers that came out of a study about six months ago indicate that the cost for developing a drug is somewhere around $800 million. The actual cost is debatable, but I think that most people agree that it's somewhere between $500 million and $800 million to develop a drug. The process for developing a drug entails three or four key areas of development. One is at the beginning when you are identifying the biological target in the body. There is a lot of biology- and genetics-related research there.

The second phase is where you do the drug development itself, taking a potential drug molecule and under-

standing whether it's an appropriate drug and whether it actually affects or interacts with the particular target in the body that you have identified from your first phase of study.

Then you have to develop the drug and make sure that it's absorbed into the body appropriately, that it metabolizes appropriately, that there are no (or minimal) toxicity effects. Then you go into a series of clinical trials, starting with trials on animals, then proceeding into trials on human subjects.

The whole process is somewhere around 12 years on average. When you take into account the expense and the number of years of development, there's a tremendous incentive to decrease the cost, even if by a factor of 10%. Ten percent is a tremendous savings for these pharmaceutical companies when you take into account the overall time and cost.

One of the advantages to the framework is that we've developed in a very modular fashion. If you can develop this on a departmental level and get immediate results with it, you can scale that to a larger group level and you can look at it from one location and from an enterprise worldwide implementation. The investment depends on the amount of time and the amount of resources that are needed to help solve some of these problems. 🌐

"One of the advantages to the framework is that we've developed in a very modular fashion"

Minimize the work

Creating Message-Based Web Services with WebSphere Studio Application Developer PART 1

BY GREG **FLURRY**

WebSphere Studio Application Developer (WSAD) includes support for developing SOAP-based Web services. For example, the WSAD Web Services wizard allows you to turn a JavaBean into a SOAP RPC–based Web service with almost no work. In addition, WSAD can create a proxy for the RPC-based Web service, greatly simplifying its use.

ABOUT THE AUTHOR

Greg Flurry is a member of the IBM Software Group Emerging Technologies area. His responsibilities include introducing Web services techologies to the IBM WebSphere product family.

E-MAIL

flurry@us.ibm.com

Some applications and services, such as UDDI, require the lower level of control offered by the SOAP message–based model, upon which the RPC-based model is built. Although WSAD doesn't yet provide automated support for SOAP message–based Web services, it's relatively easy to provide manual support in WSAD. This article shows you how to create and use message-based Web services in WSAD.

There are three steps to creating a message-based Web service in WSAD:
1. Create an RPC-based Web service to automatically create artifacts necessary for describing and deploying the Web service.
2. Implement the actual message-based service, and modify the Web service artifacts accordingly.
3. Create a client proxy for the message-based service (optional).

Creating a SOAP RPC Web Service

Using the WSAD Web Services wizard to create an RPC-based Web service minimizes the work involved in creating and deploying the message-based Web service. Since the resulting RPC-based Web service won't be used, it's best to start with a very simple Java class. For this exercise, create a Web project, create a package named "message" in the source folder of the Web project, and then create the bean described in the following code.

```
package message;

public class MessageService
{

    public String
process(String in) {
        return in;
    }
}
```

Modify the signature of the MessageService class and add the functional part of the Web service that implements the message-based Web service. Listing 1 shows the new Web service implementation.

The process() method first gets the body from the SOAP envelope received. The method extracts the request element from the body. At this point in a real service, the class would process the request element, execute the business logic required by the request, and formulate a response (in the form of a DOM Element). This simple example just returns the request.

Next, modify the Web service deployment descriptor (dds.xml) produced by the Web Services wizard. The dds.xml file can be found in the webApplication folder of the Web project containing the Web service. Simply add the attribute type="message" to the <isd:service> element in the deployment descriptor as shown in Listing 2.

After saving the modified deployment descriptor, restart the application server to deploy the Web service and make it available to clients.

Creating a SOAP Message Client Proxy

Creating a proxy for the Web service isn't absolutely necessary, but it simplifies client access to the Web service. Create another package in the source folder of the Web project; this example uses "proxy.soap.message" for the package name. The proxy itself is shown in Listing 3.

The send() method sends the request, in this case in the form of a DOM Element, to the Web service. The send() method uses the createBody() method to wrap the request in a body element with the correct name ("process") and namespace attribute. The "xmlns" attribute identifies the Web service on the server; it should be set equal to the "id" attribute in the <isd:service> element in the deployment descriptor generated by WSAD. The receive() method allows a client to retrieve the response produced by the Web service.

This generic proxy is useful for sending and receiving "free-form" documents because it accepts and returns a DOM Element. It's possible to construct a proxy with a send() method that takes one or more parameters of a specific Java type, e.g., a String, int, etc. Likewise, it's possible to have the receive() method return a specific Java type. A proxy constructed in this fashion must construct the DOM Element sent to the

Expand the Web project that contains the MessageService class, select the class, right click, and select New–>Other. On the left side of the resulting dialog, select Web Service on the right side and then select Next to start the Web Services wizard. In the initial panel of the wizard (see Figure 1), uncheck Generate Proxy, uncheck Start Web Service, and then select Finish. The wizard creates the WSDL describing the RPC-based Web service as well as a deployment descriptor. The wizard will also start the application server in which the Web service is deployed.

SOAP Message–Based Web Services

While SOAP RPC Web services can have virtually any signature with multiple operations, SOAP message–based Web services in the Apache SOAP 2.2 implementation included with WSAD all have exactly the same signature, with a single operation, shown in the following code. See the References section for more information.

```
public class
<SOAPMessageService> {

    public void process(
        Envelope env,
        SOAPContext
requestContext,
        SOAPContext
responseContext) {
        ...
    }
}
```

Modify the signature of the MessageService class and add the functional part of the Web service that implements the message-based Web service. Listing 1 shows the new Web service implementation.

service and parse the DOM Element received from the service.

Creating a SOAP Message Client

Listing 4 shows a client that uses the generalized proxy described above. The best place to create the client is in the default package of the Web project containing the Web service and the proxy.

Since the proxy requires an element as input, the client first creates a simple element named "test" with one attribute named "myAtt". The client then sets the URL for the service implementation; the value should be set to the value of the location attribute of the <service/port/soap:address> element in the "service" WSDL file generated by WSAD, except that you must substitute "messagerouter" for the "rpcrouter" in the attribute value. The client then sends the request to the service. Since the proxy for the message Web service models an asynchronous operation, the client can now do other things before retrieving the reply. Eventually, the client receives the response from the Web service. This example simply prints the attribute, which should match the attribute in the request. When you run the test client shown in Listing 4, you'll get the following result in the WSAD Console view:

The web service echoed the attribute; the value is "good"

This is exactly what's expected.

Digging Deeper

It's instructive to dig deeper into the operation of the message Web service to discover what's really happening. One way to do this is by examining the SOAP messages flowing between the client and the Web service. WSAD provides the TCP/IP Monitor that can be used for this; see the WSAD Help for more information about the monitor and how to get it running. The application server providing the Web service by default listens on port 8080. Configure the monitor to listen on port 8081 and pass requests to port 8080, then start
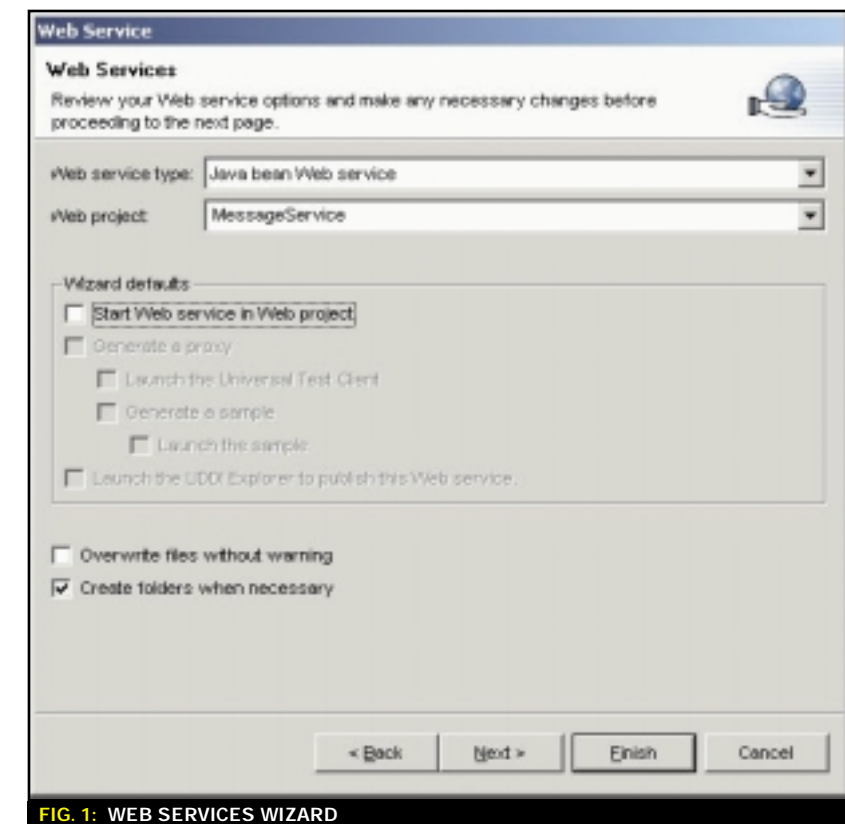

FIG. 1: WEB SERVICES WIZARD

the monitor. The only change necessary in the test client is to change "8080" in the setURL() call to "8081". Do this and run the client again. You'll see the same result in the Console view, but the monitor captures the request and response so you can examine them in the TCP/IP Monitor view.

Listing 5 shows the request sent by the client proxy. You can see that the body of the request contains the <process> element, which in turn contains the <test> element with its attribute.

Listing 6 shows the response sent as a result of the send() method. The response contains the <test> element with its attribute. You may be surprised to find that the response comes back as a result of the send() method. You can confirm this by putting a breakpoint in the test client before calling the receive() method

and running the test client in debug mode.

Summary

This article walked you through the steps necessary to develop message-based Web services with WebSphere Studio Application Developer. While not as well supported as the more widely used RPC-based Web services, this type of Web service can be inplemented and tested with WSAD.

One final note: the intentional split of the send() and receive() methods in the proxy allows a client using the proxy to model asynchronous operations where the send() method starts the Web service, but does not wait for it to finish; instead, the receive() method actually waits for the Web service to finish. In reality, the Web service is synchronous because the Apache SOAP Message.send() method invoked in

the proxy is synchronous, so the repsonse actually comes back as a result of the proxy send() method. This means that this client can't operate in an asynchronous manner; it can't do anything until the proxy send() method returns. To really be asynchronous, the proxy must implement some multithreading, which I'll cover in Part 2.

References
- **Apache SOAP version v2.3.1:** http://xml.apache.org/soap/docs/index.html
- **UDDI:** http://uddi.org
- **Web services zone on IBM developerWorks:** www.ibm.com/developerworks/webservices
- **IBM developerWorks tutorial, Creating a complete Web service:** www.ibm.com/developerworks/training 🌐

LISTING 1
```
package message;

import org.apache.soap.*;
import org.apache.soap.rpc.SOAPContext;
import org.apache.xml.serialize.*;
import org.w3c.dom.*;
import java.io.*;

public class MessageService {

    public void process(
        Envelope env,
        SOAPContext requestContext,
        SOAPContext responseContext) {

        Document doc;
        Node bodyNode;
        Node bodyNodeParent = null;

        try {
            bodyNode = (Node)
            env.getBody().getBodyEntries()
            .elementAt(0);
            Element request = (Element)
            bodyNode.getFirstChild();

            // make use of existing DOM object structure
            and
            // just replace the body to return response
            Node bodyNodeParent = bodyNode.getParentNode();
            bodyNodeParent.removeChild(bodyNode);

            // add the response element to the XML doc
```

```
            doc = bodyNode.getOwnerDocument();

            Node returnNode = doc.importNode(request, true);
            bodyNodeParent.appendChild(returnNode);

            // convert DOM structure to XML
            OutputFormat format = new OutputFormat(doc);
            StringWriter response = new StringWriter();
            XMLSerializer serial = new
            XMLSerializer(response, format);
            serial.asDOMSerializer();
            serial.serialize((Element)
            doc.getDocumentElement());
            responseContext.setRootPart(response.toString(),
            "text/xml");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

LISTING 2
```
<root>
<isd:service xmlns:isd="http://xml.apache.org/xml-
soap/deployment"
    id="http://tempuri.org/message.MessageService"
    type="message"
    checkMustUnderstands="false">
  <isd:provider type="java" scope="Application"
  methods="process">
    <isd:java class="message.MessageService"
    static="false"/>
  </isd:provider>
</isd:service>
</root>
```


Chart Your Course
to Success in IT...
...order your copy of **Java Trends: 2003**

EnginData's research is invaluable to leading IT vendors, integrators, Fortune 500 companies, trade-show organizers, publishers, and e-business organizations worldwide.

Don't go astray. In the vast sea of Internet technology, market conditions change constantly. Will Java remain the hot platform it is today? Will C# rapidly gain ground? What are the world's foremost Java developers aiming their sights toward? Which companies come to mind when planning their next project? How can their thinking direct your efforts?
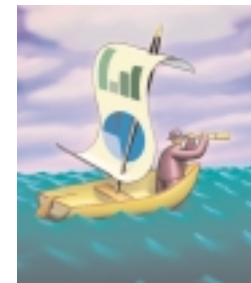
EnginData Research has studied the IT industry extensively, spotting many key trends and alerting industry leaders along the way. In the first quarter of 2002, they embarked on their most challenging mission ever: the most in-depth study ever done of the Java development marketplace.

After months of analyzing and cross-referencing more than 10,500 comprehensive questionnaires, the results are now available.
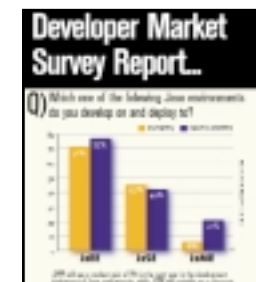
Here's just a sample of the critical data points the Java report delivers...
- ✔ Current and projected usage of languages and platforms
- ✔ Multiple rankings of hundreds of software vendors and tools
- ✔ Types of applications being developed
- ✔ Databases being used
- ✔ Purchasing patterns
- ✔ Company size
- ✔ Development and purchasing timelines
- ✔ Perceptions and usage of Web services
- ✔ Preferred Java IDE
- ✔ J2EE, J2ME, J2SE usage comparisons

As an IT specialist, **EnginData Research** focuses exclusively on three leading drivers in IT today – Java, Web services, and wireless development. Coming soon, our Web services survey will deliver such critical knowledge as:
- ✔ Time frame for developing Web services – enabled apps
- ✔ Percentage of apps with Web services today
- ✔ Sourcing and hosting Web services
- ✔ Perceived leaders in Web services tools and solutions

Navigate your company through the challenging IT marketplace with the trusted knowledge and intelligence of **EnginData Research**. Order your copy of the 2002–2003 Java market study by calling Margie Downs at 201-802-3082, or visiting our Web site.

engindata RESEARCH

www.engindata.com

```
LISTING 3

package proxy.soap.message;


import java.net.*;
import java.util.*;
import org.w3c.dom.*;
import org.apache.soap.*;
import org.apache.soap.messaging.Message;


public class MessageServiceProxy {

 protected Message soapMessage;
    protected URL theURL = null;

    public MessageServiceProxy() {
        soapMessage = new Message();
        soapMessage.setSOAPTransport(null);
    }

    public synchronized void setURL(String url) throws
    Exception {
        theURL = new URL(url);
    }

    public void send(Element payload) throws SOAPException
{
        Element body = createBody(payload);
        Envelope sendEnv = new Envelope();
        Body sendBody = new Body();
        Vector bodyEntries = new Vector();
        bodyEntries.add(body);
        sendBody.setBodyEntries(bodyEntries);
        sendEnv.setBody(sendBody);
        soapMessage.send(theURL, "", sendEnv);
    }


    public Element receive() throws SOAPException {
        Element response = null;
        Envelope responseEnv =
        soapMessage.receiveEnvelope();
        Body responseBody = responseEnv.getBody();
        response = (Element)
        responseBody.getBodyEntries().firstElement();
        return response;
    }


    protected Element createBody(Element payload) {
        Element body = null;
        try {
            Document doc = payload.getOwnerDocument();
            body = doc.createElement("process");
            body.setAttribute("xmlns",
            "http://tempuri.org/message.MessageService");
            body.appendChild(payload);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        return body;
    }
}


LISTING 4

import message.*;
import proxy.soap.message.*;
import org.w3c.dom.*;
import org.apache.xerces.dom.*;

public class TestMes {
```

```
    public static void main(String[] args) {


    MessageServiceProxy proxy = new
    MessageServiceProxy();
        try {
            DocumentImpl doc = new DocumentImpl();
            Element inputEl = doc.createElement("test");
            inputEl.setAttribute("myAtt", "good");


proxy.setURL("http://localhost:8080/MessageService/servlet/me
ssagerouter");
            proxy.send(inputEl);

            Element response = proxy.receive();
            String result = response.getAttribute("myAtt");
            System.out.println("The web service echoed the
            attribute;
                the value is \""+ result + "\"");

        } catch (Exception ex) {
          ex.printStackTrace();
        }
    }
}


LISTING 5

POST /MessageService/servlet/messagerouter HTTP/1.0
Host: localhost:8080
Content-Type: text/xml; charset=utf-8
Content-Length: 367
SOAPAction: ""

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <process xmlns="http://tempuri.org/message.
    MessageService">
      <test myAtt="good"/>
    </process>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>


LISTING 6

HTTP/1.1 200 OK
Server: WebSphere Application Server/4.0
Content-Type: text/xml
Set-Cookie: JSESSIONID=0000BHWUPBKL2DOVAPKAOXTQWAQ:-1;Path=/
Cache-Control: no-cache="set-cookie,set-cookie2"
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Length: 291
Content-Language: en
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <test myAtt="good"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

TOOLS

TOOLS

Simplifying portlet development

Portal Toolkit for WebSphere Portal

BY DAVID B. LECTION

On April 9, 2002, IBM announced WebSphere Portal for Multiplatforms

version 4.1, a robust third-generation product that enables customers

to build high-performance portal-based solutions.

There are a myriad of new features in the new portal offering:
- **J2EE compliance**
- **Improved API:** Portlets are now further specializations of servlets
- **Improved portlet-rendering performance:** Includes parallel rendering of portlets
- **Improved security model:** Includes improvements in single sign-on functions and password vaulting
- **New toolkit for portlet development**

WebSphere Portal for Multiplatforms Packages

Three packaged versions of the portal are available. The base package, WebSphere Portal Enable, features WebSphere Portal, WebSphere Personalization, Web Content Publisher, and WebSphere Studio Application Developer (WSAD).

WebSphere Portal Extend, the mid-level package, includes all the features of the Enable package plus the basic suite of Lotus collaboration tools. These advanced Lotus tools include instant messaging and access to team rooms; Lotus Collaborative Places, which adds place awareness to any portlet; team collaborations; and shared team documents. Also included is Lotus Domino Extended Search, a tool for searching across extended

data stores, including relational databases, Domino databases, and World Wide Web search engines. The Extend package also includes Site Analyzer, a tool for monitoring the performance of the portal server.

The premier portal package, WebSphere Portal Experience, gives enterprises the most complete set of tools available to build a high-performance portal solution. This package includes all the features of the Enable and Extend packages and adds the IBM Tivoli Access Manager, IBM Content Manager, and IBM Enterprise Information Portal. These tools give the enterprise industrial-strength content and data-management tools integrated with the portal.

WebSphere Studio Application Developer

Each package includes a license for WSAD, which integrates several Web application development roles into one tool that provides best-of-breed, integrated development support for:
- Building J2EE applications with HTML pages, servlets, JavaServer Page (JSP) files, and Enterprise JavaBean (EJB) components
- Creating Web applications based on Web services open standards
- Creating and modifying XML documents from DTDs and XML schemas

- Visual WYSIWYG creation of HTML content documents and JSPs
- Enabling a collaborative team environment for development of applications
- Tools for optimizing application performance
- Enabling end-to-end local and remote application testing
- Increasing productivity and creating high-quality applications using wizards, code generators, and best practices

Eclipse: A Platform for Tools Development

WSAD is built on the Eclipse platform, an open-source framework for the development of tools for an interactive development environment (IDE). The Eclipse platform supports:
- Construction of a variety of tools for application development
- Integration of tools from multiple tool providers, including independent software vendors
- Tools to manipulate arbitrary content types (e.g., HTML, Java, C, JSP, EJB, XML, and GIF)
- Seamless integration of tools within and across different content types and tools providers
- Both GUI- and non-GUI–based application development environments
- A wide range of operating systems, including Microsoft Windows and Linux

In addition, the Eclipse platform capitalizes on the popularity of the Java programming language for writing tools. Its principal role is to provide tool providers with mechanisms to use and rules to follow that lead to seamlessly integrated tools. These mechanisms are exposed via well-defined API interfaces, classes, and methods.

The Eclipse workbench (see Figure 1) features a project-based paradigm. As a project is developed, project components are created and modified using tools built into the platform as plug-ins.

When you open or create an Eclipse project, the appropriate set of tools and viewers is opened in the workbench. This placement and view of the tools is called a development perspective. For example, WSAD includes perspectives for Web development, J2EE application development, and generic Java development. When you open help in the workbench, the help perspective is opened. You can change the current perspective at any time.

Introducing the Portal Toolkit

The Portal Toolkit included with all versions of WebSphere Portal is a set of extensions that adds a palette of portlet development tools to WSAD. These tools integrate seamlessly with WSAD and enable rapid development of WebSphere Portal portlet applications written in Java. (A portlet application is a set of portlets packaged together in a Web Archive, or WAR, file.)

PORTLET APPLICATION BASE PROJECT

The portlet application base project extends the J2EE Web application project and packaging mechanism for portlet applications. The base project includes a base Web application

descriptor document, web.xml, and a portlet application descriptor document, portlet.xml. A portlet application is packaged in a WAR file with these documents in the WEB-INF directory of the application's WAR file. As portlets are added to a project, the names of the portlets and other settings are automatically added to both the Web application descriptor and the portlet application descriptor.

PORTLET AND WEB APPLICATION DESCRIPTOR WIZARDS

The base project also includes creation and management wizards for both the portlet application descriptor and the Web application descriptor. Manual development and maintenance of these descriptor documents is a complex task – values and settings in the Web application descriptor are referenced in the portlet application descriptor and must be kept in sync. The wizards make the management of these descriptor documents an easy task. As values in the Web application descriptor are changed, they're automatically updated in the portlet application descriptor. Users can also update and tweak the values in these documents in an XML editor view, with the new values reflected in the wizards.

PORTLET APPLICATION CREATION WIZARD

The portlet application creation wizard empowers the user to create complete portlet application drafts. A draft version of a portlet application is ready for the addition of business logic. Simply answer a few questions in the wizard and specify pertinent portlet options. The wizard then generates a complete portlet application draft based on one of the available portlet helper classes: PortletAdapter, MVCPortlet, JSPPortlet, ServletInvokerPortlet, or XSLPortlet. You add business logic and other enhancements to the draft, completing the final portlet application right in the tool. It's now ready for debugging or deployment to a portal server.

PORTLET APPLICATION SAMPLES

Several portlet application samples are included in the toolkit. These samples are complete portlet applications that demonstrate a particular function. For example, the Yourco White Pages portlet application demonstrates the use and access of a database within a portlet.

ONE-STEP PORTLET DEBUGGER

Perhaps the most powerful feature of the Portal Toolkit is the portlet debugger. The debugger supports one-step debugging. When you select the "Run On Server" function on a portlet application, in a single step the application is automatically exported and installed on a target portal server. The portlets in the portlet application are then added to a portal debug page. Finally, a browser is opened and a URL is sent to the browser to log the user into the portal and display the debug page for the user. In previous versions of the portal, testing a portlet was a manual process with a dozen steps – now you can debug a portlet application with a single mouse click.

The portlet debugger has a very robust set of features. Breakpoints can be set anywhere in a portlet application's Java code, or in a JSP. When a breakpoint is reached, the debugger stops and shows the breakpoint in a source view of the portlet. Values in variables and objects may be displayed.

FIG. 1: ECLIPSE WORKBENCH
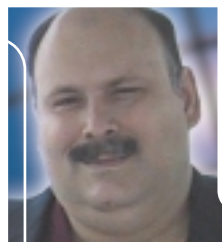
ABOUT THE AUTHOR
David B. Lection is a senior software engineer at IBM Corporation. David is the lead development architect for the WebSphere Everyplace Access and Portal Toolkits.
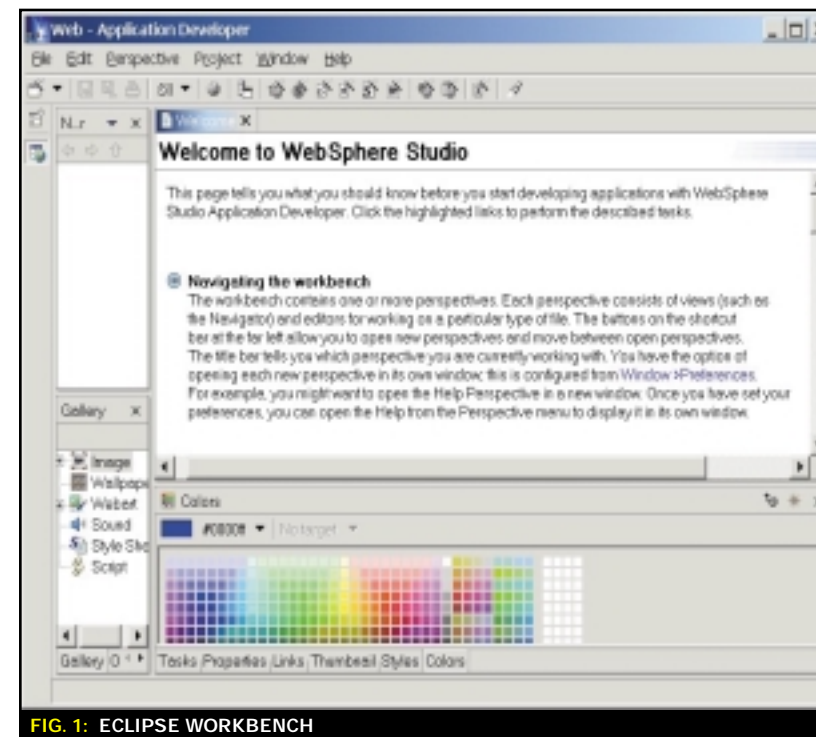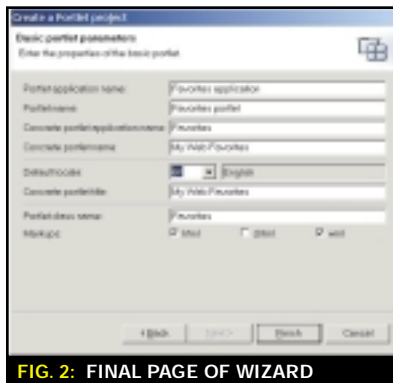
E-MAIL
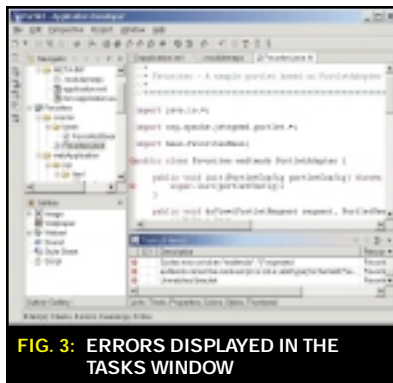lection@us.ibm.com

FIG. 2: FINAL PAGE OF WIZARD

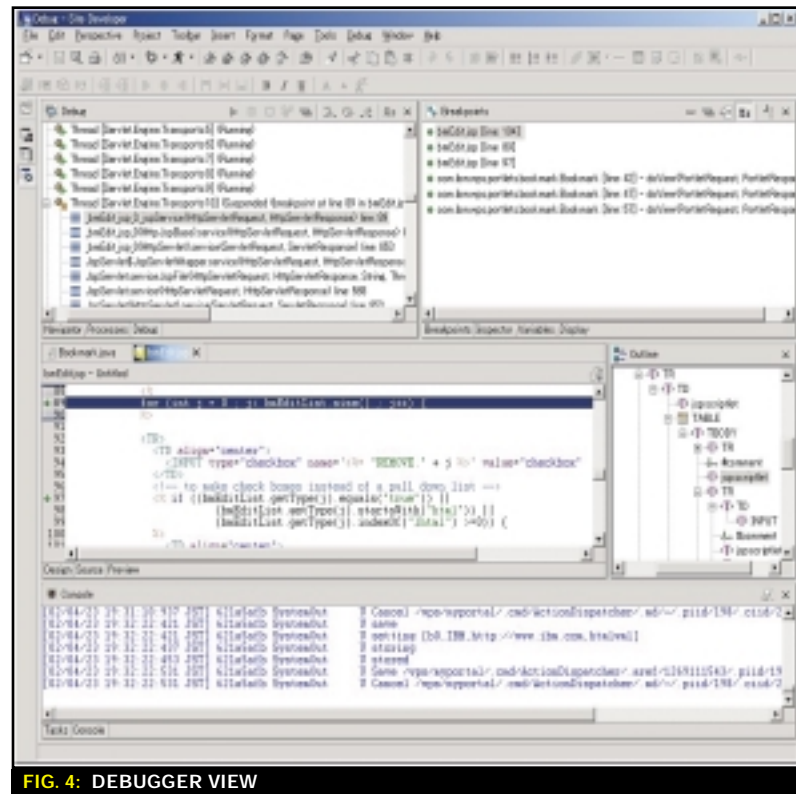**FIG. 3:** ERRORS DISPLAYED IN THE TASKS WINDOW

FIG. 4: DEBUGGER VIEW

SERVER CONFIGURATION

Server configuration allows the user to create a remote portal server configuration. This configuration is used to determine the target portal server and associated portal parameters used for debugging.

Developing Portlets with the Portal Toolkit

Developing portlet applications is easy. Using the project wizards, samples, and debugger you can build a custom portlet application in minutes, not hours.

Basic development of a portlet application involves four phases: creating the portlet application draft, adding custom logic and views to the application, debugging the application, and deploying it to a production portal server.

We'll walk through these phases of development and show you how to develop a Favorites portlet. The Favorites portlet saves and displays a list of the user's favorite URLs.

CREATING A DRAFT VERSION OF A PORTLET APPLICATION

The first step in development is to create an initial version of a portlet

FIG. 5: ACTIVATING THE APPLICATION

application. The Favorites portlet will be based on a subclass of the portal's PortletAdapter helper class. Portlets that subclass the PortletAdapter class can support all four portlet display modes. The draft application includes sample JSPs for each display mode.

To create the portlet application draft, click on File | New | Portlet Application Development. A dialog is presented that allows you to enter the name of the portlet application, in this case, "Favorites." The project name and portlet application root directory are known as the context root in the WSAD workspace. On page

two of the wizard the type of portlet is selected.

On the final page of the wizard, the portlet application name and the portlet name are selected (see Figure 2). The display markups the portlet is required to support are also specified.

When the finish button is pressed, a complete portlet application skeleton is generated. This skeleton application includes the portlet and Web descriptor documents, the portlet Java class, initial display JSPs for each markup, and the associated JavaBeans passed to the display JSPs.

ADDING THE BUSINESS LOGIC TO THE PORTLET

Once the initial portlet application has been generated, add the business logic to the portlets in the portlet application. Business logic is typically added in the base portlet Java classes. As you make changes to the Java source statements and save the files, the portlet development environment automatically compiles the application and displays any errors in the Tasks window (see Figure 3).

The portlet development environment also provides source editors for

JSPs that are part of the portlet application. In the generated Favorites application, both the HTML and WML markups were specified as being supported by the application. Sample JSPs for each of these markups, for all the portlet display modes, are created as part of the portlet application. The initial generated JSPs contain no custom user interface markup, but using the JSP editor it's simple to add custom markup to these pages.

DEBUGGING THE FAVORITES PORTLET APPLICATION

Once the business logic and all other code development have been completed, the portlet application is ready for debugging. Invoking the portlet debugger is a simple matter of selecting the portlet project and selecting "Run on Server." The portlet application is dynamically deployed to the portal server, installed, and activated. The portlets in the portlet application are then added to a debug page and a Web browser is opened,

displaying the portal debug page with the portlets.

The debug perspective is opened in the workbench. In the debugger, breakpoints may be set in Java code or JSPs. Objects and variables may inspected for particular values. The debugger provides all the features required to debug your portlet application.

In the debugger view shown in Figure 4, a breakpoint has been reached in a JSP.

DEPLOYING THE FAVORITES PORTLET APPLICATION

Once the portlet application code is complete, the application can be deployed to a production portal server. You'll need to log on to the portal server, install the portlet application, and activate the application (see Figure 5). The portlet application is now available to all portal users!

Summary

The Portal Toolkit for WebSphere Portal Version 4.1 is the first release

specifically designed for portlet application development. It is an extension package that adds on to WebSphere Studio Application Developer and provides significant, powerful tools for the creation of portlet applications, including application draft creation, HTML and JSP content creation, and portlet application debug and deploy functions.

In future releases, IBM will enhance the functions of the Portal Toolkit to further enhance developer productivity, including:

• Advanced portlet application samples
• Support for advanced portlet user interfaces
• Rapid application development (RAD) portlet application development tools

If you're developing portlet applications for WebSphere Portal, then the Portal Toolkit is a must-have in your selection of software development tools!

Authentication in WebSphere Portal

in
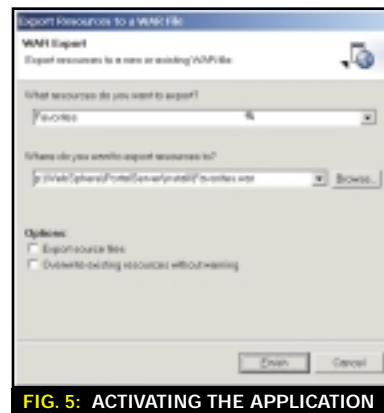
The full SSO experience

WRITTEN BY INGO **SCHUSTER**

ABOUT THE AUTHOR

Ingo Schuster is part of IBM's pervasive computing division and has been a member of the Portal Server development team in Böblingen since its first release. He holds a degree in computer science from the University of Tübingen.

E-MAIL

ingo.schuster@de.ibm.com

Enterprise application integration (EAI) is a prime objective driving the decision to implement a portal. Portals are often used to integrate data and applications from remote systems and present them in a unified manner to users through a Web-based workspace. Because these back-end systems can contain sensitive business information and functionality (for example, a company's order control system) or private data (e-mail or employee records), access should be well controlled.

The basis of access control is authentication, which ensures that the person or program requesting access is identified beyond doubt. It's recommended that back-end systems don't relinquish control of their application security to the portal server, but continue to do their own authentication and authorization. However, users shouldn't be repeatedly prompted to authenticate. This is where single sign-on (SSO) comes into play.

SSO isn't an entirely new idea. However, it raises quite a few problems, and many attempts to develop a general SSO solution have failed. Users are typically known by a variety of identities – one per system in the worst case. The identity a user proves at one system must be mapped securely to an identity that a second system can handle. In addition, in companies that have grown their back-end systems over the years, there are many systems that no one dares to touch. Therefore, it's a common requirement that they're integrated into an SSO realm without any modification to the system itself.

This is a difficult problem. However, using a portal to realize EAI gives you an advantage: the portal becomes a single entry point for all user requests. At this central instance, it's possible to map the user identities transparently to the back-end systems. All you need is a registry for those mappings and a service that leverages this feature to the portlets. WebSphere Portal calls this component the credential vault.

This article explores authentication in the context of WebSphere Portal 4.1 and demonstrates how to provide a full SSO experience to portal users.

Authentication Setups

Portal server authentication is like the front door of a building with all room doors unlocked – the key to the portal is the master key to all applications and information within it. WebSphere Portal leverages the application server's authentication subsystem, which provides support for the following authentication setups:

- **Use of native WebSphere Application Server authentication:** A custom login form posts the user's authentication data to the portal servlet that requests WebSphere Application Server security to validate the user's authentication data. This setup exploits the integration of WebSphere Portal with WebSphere Application Server and the fact that the portal can be configured as a secure Web application.
- **Use of an authentication proxy or HTTP server plug-in:** WebSphere Application Server provides the Trust Association Interceptor (TAI) interface, which allows it to establish trust with authentication proxies and authentication plug-ins.

AUTHENTICATION THROUGH WEBSPHERE APPLICATION SERVER SECURITY

To use WebSphere Application Server security, the portal is configured as a secure Web application. When WebSphere Application Server receives a request for the portal application, its security component redirects the client to a login form that prompts the user to provide authentication credentials. The form posts the credentials to WebSphere Portal's servlet, which makes the required calls to WebSphere Application Server's security functions to log the user in to WebSphere Application Server's security context. WebSphere Application Server authenticates the user by checking the provided credentials against the user registry, which can be an LDAP directory or a custom user registry.

AUTHENTICATION THROUGH PROXY OR HTTP PLUG-IN

An external authentication component can protect the portal by intercepting all requests targeted to WebSphere Portal destinations. Such an authentication component can be a proxy server like Tivoli Access Manager's WebSEAL or a plug-in to the IBM Edge Server, like WebSEAL Lite from Tivoli, SiteMinder from Netegrity, or GetAccess from Entrust.

Such an external authentication component works by checking the provided credentials against a user registry (in most cases an LDAP directory). TAIs registered with WebSphere Application Server establish trust between the server and the authentication component protecting it. This is a common scenario in production environments.

Trust Association Interceptor

With a TAI registered, WebSphere Application Server relies on an external authentication component for authenticating requests rather than performing authentication itself. The interface that WebSphere Application Server defines for a TAI has functions with which the TAI indicates that it can handle a request and that the request is already authenticated.

After a request for a portal destination passes the external authentication component, it's received by WebSphere Application Server and sequentially passed to the registered TAIs until one TAI indicates that it's responsible for authentication and either accepts or rejects the request.

If no TAI can handle the incoming request, WebSphere Application Server falls back to its native authentication, just as if there were no TAIs. In the case of WebSphere Portal, the client would be redirected to the custom login form. However, this should occur only with requests that bypassed the external authentication component.

Single Sign-On

The goal of SSO is to authenticate a user one time within an environment and use that single authentication (for the duration of the session) as a basis for access to other applications, systems, and networks.

In the context of the portal server, there are two SSO realms, the Client-to-WebApp SSO realm and the Portal-to-BackEnd SSO realm (see Figure 1).

With Client-to-WebApp SSO, a client logs in to one Web application one time and is then able to access all other applications that are part of the same SSO realm without a second authentication challenge. It doesn't matter whether
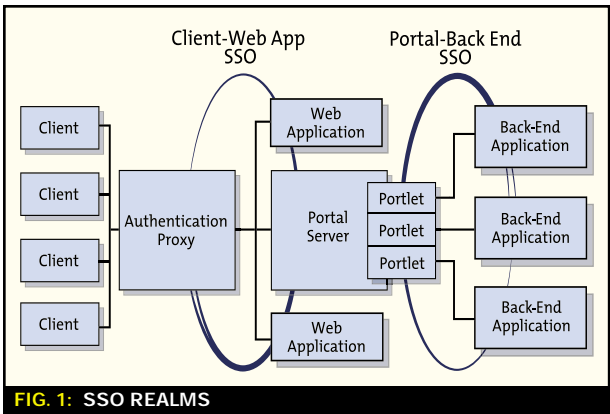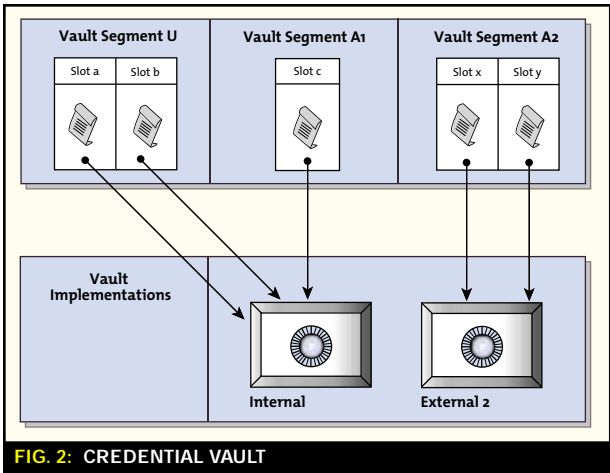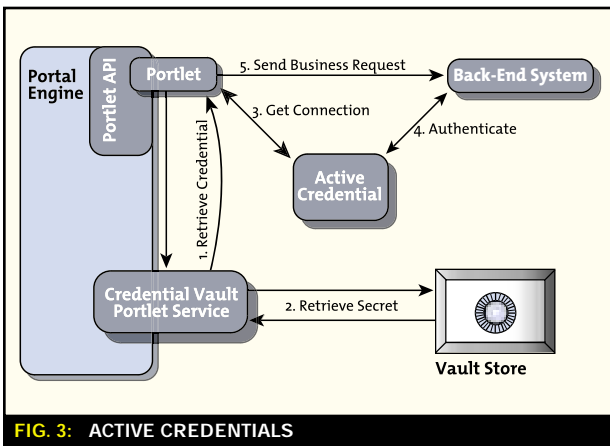
FIG. 1: SSO REALMS

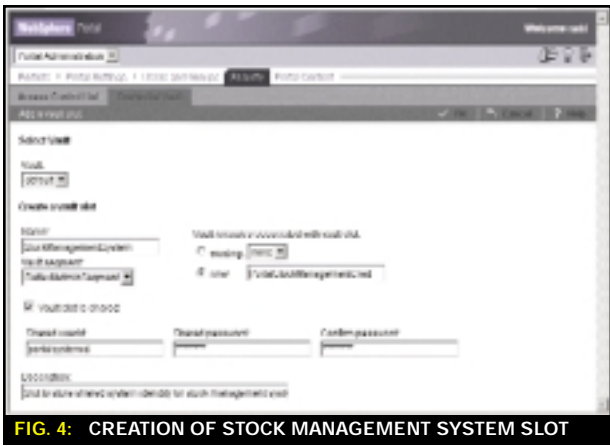FIG. 2: CREDENTIAL VAULT

FIG. 3: ACTIVE CREDENTIALS

FIG. 4: CREATION OF STOCK MANAGEMENT SYSTEM SLOT

the portal server or one of the other trusting Web applications is the authenticating Web application.

Portal-to-BackEnd SSO allows a portal client to log in to the portal once and then access a number of back-end applications without having to authenticate at each application.

CLIENT-TO-WEBAPP SSO

Client-to-WebApp SSO can be provided through a variety of mechanisms, namely:
1. Application server built-in SSO support (LTPA)
2. Authentication proxy SSO support (WebSEAL, Netegrity SiteMinder, etc.)
3. Other third-party SSO frameworks/environments, such as GetAccess from Entrust

The functional principle of these SSO solutions is essentially the same. Upon successful authentication the client gets a security token that entitles it to get through the policy enforcement point(s) (PEP) with subsequent requests. It's not important in this context whether this is a number of distributed PEPs, such as several application servers (mechanism 1); a single PEP, such as one central authentication proxy (mechanism 2); or a mixture of both (mechanism 3).

PORTAL-TO-BACKEND SSO

Starting with Version 4.1, WebSphere Portal offers a credential vault as a portlet service. This service provides a mechanism to portlets (and to the portal itself) for mapping from a user identity (usually a user ID) to another user identity and usually a credential (such as a password) for that user ID.

Portlets no longer need to store user credentials as part of the user-specific portlet data. In fact, it's recommended that such portlets be migrated so that they make use of the credential vault.

The Credential Vault Service is composed of three levels. At the lowest level are vault adapters. A vault adapter is the SPI-specific plug-in between the vault service and a specific vault implementation. WebSphere Portal provides a default implementation; you can also use the vault implementation provided by IBM Tivoli Access Manager.

At the next level are vault segments, which correspond to a section of a vault that's either user or administrator managed.

At the top level are vault slots. A vault slot corresponds to a particular resource in a vault segment that a user may apply a secret to. It's through a vault slot that user secrets are created, retrieved, modified, and deleted.

CREDENTIAL VAULT CONCEPTS

WebSphere Portal's credential vault (see Figure 2) is organized as follows:
• The *portal administrator* can partition the vault into several vault segments. Each vault segment is associated with a certain vault implementation (the actual store) that allows different credentials to be kept in different physical stores. The segments can be created and configured only by portal administrators.
• A *vault segment* contains one or more credential slots, the "drawers" where portlets store and retrieve the cre-

dentials. Each slot holds one credential (or one per user if it's an admin-defined slot).
• A *credential slot* is linked to a resource in a vault implementation, where the credential secrets are actually stored. Examples for vault implementations are the portal server's default database vault or Tivoli Access Manager.

A vault segment is flagged to be either admin managed or user managed. While portlets (on behalf of a portal user) can set and retrieve credentials in both types of segments, they're permitted to create credential slots only in user-managed vault segments. Therefore, portlets that need a credential to complete their service have basically two options:
1. Use an existing credential slot that's been defined by the portal administrator in an admin-managed vault segment.
2. Create an "own" credential slot in a user-managed vault segment.

Which variant to choose depends on the use of the portlets in the portal installation.

The credential vault distinguishes between three different types of credential slots:
• *Shared-system credential slot:* Stores system credentials, i.e., credentials where the actual secret is shared among all users and portlets
• *Portlet-shared credential slot:* Stores a user credential that's shared among a user's portlets, i.e., the

secret is user specific but the same for all the user's portlets.
• *Portlet-private slot:* Stores a user credential that's not shared among portlets, i.e., the credential secret is user specific as well as specific to a certain concrete portlet instance.

CREDENTIAL VAULT FUNCTION

The Credential Vault Service maps a requested credential slot, a user ID, and a portlet ID to a resource in the vault; portlets can only retrieve a credential if a respective mapping rule exists. Some credentials (secrets) will be stored and managed by the portal (local default vault store); secrets that aren't stored locally will be acquired from the respective external vault.

If a credential isn't available, a respective exception is thrown. The portlet service passes this exception to the portlet to allow appropriate error-handling (e.g., ask the user to set the credential through the portlet's edit mode). The credential vault won't allow any person other than the credential owner to manage and/or use the credentials – not even a portal administrator. Portal administrators can configure the credential services that aren't controlled by the user. This includes the management of the vault segments, admin-defined slots, and system (shared) credentials. An administrator may not administer the user-defined credential slots. (Of course, all user-defined slots will be deleted if the respective user is deleted.)

The portal end user manages credentials directly in the portlet's edit mode. With WebSphere Portal 4, it's the portlet's responsibility to provide all necessary functions that allow the user to manage the slots used by this portlet. IBM provides framework sample code that supports portlet developers implementing these credential-management functions.

The portal engine, however, provides all the interfaces necessary to write a general credential management portlet for portal end users. Future versions of WebSphere Portal might ship with such a portlet.

Usually, a portlet binds the credentials it needs to certain credential slots only at runtime, not during deployment. However, it may predefine default values in its PortletSettings to ease the administrator's work or to support a staged development/test environment.

ACTIVE CREDENTIALS

Just like real valuables, credentials are perfectly safe as long as they're stored in the vault. However, when they're taken out, the owner might worry about their safety: What if the portlet developer didn't pay enough attention and violated security rules? WebSphere Portal's answer to this problem is *active credentials.*

The Credential Vault Service returns credentials as credential objects. An active credential object hides the credential's secret from the portlet; there's no method to extract it. In return, active credential objects offer business methods that take care of the authentication at the backend system. That way, a portlet can trigger remote authentication without even knowing the credential's secret; the credential object provides the portlet with an already-authenticated connection. From a security point of view the portlets never get in touch with the credential secrets, and a portlet developer can't accidentally violate security rules (see Figure 3).

All credential types available within the portal are registered in a credential-type registry. WebSphere Portal Server 4.1 provides a small set of credential types out of the box; additional credential objects can easily be added. The active credential classes shipped with WebSphere Portal provide support for standard authentication mechanisms such as HttpBasicAuth, HttpFormBasedAuth, or POP3 authentication. Furthermore, there are credentials that integrate with common SSO solutions, namely LTPA, WebSEAL, and SiteMinder.

Programming Example

Scenario: A company's buying department runs a portal that integrates different legacy applications. One is an ordering-server application with a simple Web front end. Several employees use the ordering portlet, but the application is secured by a system ID – it doesn't support multiple user accounts.

Design solution: The portlet needs to access the legacy application under a system ID. The portal administrator therefore creates a credential slot, "StockManagementSystem", in an admin-managed vault segment; marks it as a shared system credential; and stores the ordering system ID and password in this slot (see Figure 4).

The portlet allows the credential slot ID to be set during deployment, so the buying department's employees don't have to care about credentials at all; they can enjoy an SSO experience right from the start. In the portlet's code, the credential service is used as follows:

1. Get the Credential Vault Service.

```
PortletContext context = this.getPortlet
  Config().getContext();
CredentialVaultService service =
  (CredentialVaultService)
  context.getService(CredentialVault
  Service.class);
```

2. Retrieve the configured slot ID.

```
String slotId = (String) request.get
  Data().getAttribute("credential slot id");
```

3. Set the credential object configuration.

```
HashMap config = new HashMap();
config.put(HttpFormBasedAuthCredential
  .KEY_USERID_ATTRIBUTE_NAME,    "userid");
config.put(HttpFormBasedAuthCredential
  .KEY_PASSWORD_ATTRIBUTE_NAME, "password");
config.put( HttpFormBasedAuthCredential
  .KEY_LOGIN_URL, "OrderApp.yourco.com/login.jsp");
config.put(HttpFormBasedAuthCredential.KEY_
  LOGOUT_URL,"OrderApp.yourco.com/quit.jsp");
config.put( HttpFormBasedAuthCredential
  .KEY_USE_AUTH_COOKIES, Boolean.TRUE);
```

4. Retrieve the credential object from the credential vault.

```
HttpFormBasedAuthCredential credential =
  (HttpFormBaseduthCredential)
  service.getCredential(slotId,
  "HttpFormBasedAuth",config,request);
```

5. Use the credential object to log in at the server application, get a connection, and send the request.

```
credential.login();
connection = credential.getAuthenticated
Connection("OrderApp.yourco.com/request.jsp");

// Work with the connection: send an HTTP GET or POST and
   evaluate the response
```

6. Use the credential object to log out at the back-end application.

```
credential.logout();
```

Summary

Authentication and single sign-on are important issues for portals. WebSphere Portal Server can be set up to rely on the application server's authentication or to use an external authentication component. Existing mechanisms for client-to-Web application single sign-on are leveraged, and the portal server's credential vault creates new possibilities for seamless integration with back-end systems.

J.D. EDWARDS SELECTS PRECISE TO OPTIMIZE APPLICATIONS

(Westwood, MA) – J.D. Edwards & Company, a provider of supply chain management and CRM and ERP software and solutions, has selected Precise/Indepth for WebSphere, from Precise Software Solutions, to help ensure that its OneWorld enterprise and CRM software applications are optimized for performance when deployed at customer sites.

J.D. Edwards reviewed a variety of products and determined that Precise/Indepth for WebSphere delivered clear advantages. These include comprehensive J2EE instrumentation, correlation of J2EE performance metrics, in-depth drill-down analysis to isolate root cause, and low overhead while collecting profile data. They particularly liked the product's ability to automatically instrument their application and correlate critical performance metrics, allowing them to quickly identify performance bottlenecks. www.precise.com, www.jdedwards.com

ECLIPSE WELCOMES SIX NEW MEMBERS

(Ottawa) – The Eclipse community has welcomed six new member companies, including executives from each that now serve on the Eclipse Board of Stewards. With over 1.8 million download requests from users in over 125 countries, community interest in the open-source Eclipse project that creates the Eclipse universal tools platform for multilanguage, multi-platform and multivendor deployment continues to grow and build momentum.

The new members include Hitachi, Ltd.; Instantiations, Inc.; MontaVista Software; Scapa Technologies Limited; Telelogic; and Trans-Enterprise Integration Corp. Eclipse offers a plug-in based framework that makes it easier to create, integrate, and use software tools, saving time and money. www.eclipse.org

IBM, INVENSYS MARKET SUPPLY CHAIN SOLUTIONS

(Lake Forest, CA) – IBM Corp. and Invensys have announced they will work together to deliver solutions that streamline and optimize the manufacturing supply chain through application interoperability.

Invensys Supply Chain Accelerator solutions will leverage the experience of IBM Global Services and IBM's WebSphere infrastructure technology to integrate the flow of material and information through discrete and hybrid manufacturing enterprises. These industry solutions extend business processes and expand the capabilities of a manufacturing company's existing supply chain. www.invensys.com, www.ibm.com

HOUSEHOLD INTERNATIONAL SELECTS RESONATE

(Sunnyvale, CA) – Resonate Inc., provider of service-level management solutions, has announced that Household International, the consumer lender, has selected Resonate Commander Solutions to provide enhanced end-user experience for its business-critical applications. Household International offers a broad set of financial products and services to more than 48 million people in the United States, Canada, and the United Kingdom. Resonate Commander Solutions software is the only service-level management solution to significantly reduce problem resolution time while simultaneously routing end users around problems, ensuring transaction completion.

Household uses Resonate Commander Solutions to capture information about application, system, and network health of the infrastructure, including IBM WebSphere Application Server and IBM WebSphere MQ and correlate it with historical data and end-user service levels to proactively prevent performance or availability problems. Household is also using Resonate Central Dispatch to take real-time, corrective action to ensure that client requests are processed quickly and accurately. Resonate's approach allows for the quickest problem identification, diagnosis, and resolution, resulting in peak end-user service levels and operating efficiency. www.resonate.com

WINTERGREEN RESEARCH NAMES MQSOFTWARE #1 MARKET LEADER

(Minneapolis) – MQSoftware, Inc., a leading provider of middleware technology solutions for the IBM WebSphere family of products, has been named the number one market leader in application integration system management according to a new market research report published by WinterGreen Research.

The report, "Message-Oriented Middleware Software Market Strategies, Market Opportunities and Market Forecasts 2002 to 2007," gives a detailed analysis of the various segments, vendors, and issues that make up the middleware market. MQSoftware was named the market leader in systems to visualize and manage IBM MQSI. In the same report, MQSoftware also captured the third leading market position for the middleware messaging management segment. For the full report, go to www.wintergreenresearch.com. www.mqsoftware.com

Palm and IBM to Deliver Wireless Solutions for the Enterprise

(Santa Clara, CA and Armonk, NY) – Palm, Inc., and IBM have announced plans to develop a joint capability that will let customers securely access enterprise applications and data using select models of wireless-enabled Palm branded handheld computers, including the Palm m515 color handheld.

Through IBM's WebSphere Everyplace Access (WEA), customers would be able to build upon their technology investments and existing development skills to extend applications beyond the firewall on wired networks using wireless technology and Palm handhelds. In addition, IBM and Palm plan to deliver the first enterprise-class instant messaging and awareness solution for Palm handhelds using Lotus Sametime technology. Users of select wireless-enabled Palm handhelds would be able to exchange instant messages among themselves and with desktop Sametime users. www.palm.com
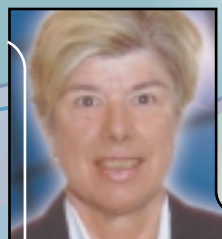
SIMPLEX KNOWLEDGE COMPANY

WWW.SKC.COM/GUIDE

THE LIFE SCIENCES REVOLUTION

Technology is changing drug research and healthcare

ABOUT THE AUTHOR

Barbara Burian directs the internal marketing communications and knowledge management programs within IBM Life Sciences Solutions. In her 17-year career with IBM, she has held a variety of positions in sales and services. Before joining IBM, she worked in medical research at Lederle Laboratories and New York University School of Medicine. Mrs. Burian has served on the faculties of New York University and Fairfield University.

E-MAIL
bjburian@us.ibm.com

WRITTEN BY
BARBARA **BURIAN**

I n the past few years we've seen a revolution taking place in the life sciences, a revolution driven by information technology, which has become the engine of experimental biology. As a result, we are seeing the convergence of information technology and the biological sciences, a convergence that will affect the way we live and that holds the potential to greatly improve the quality and effectiveness of medical care.

Perhaps the term "life sciences" is new to you. The life sciences market includes the pharmaceutical, biotechnology, genomic, e-health, and agri-science industries. This market is essentially dedicated to drug discovery, to understanding genomics, proteomics, structural biology, and emerging areas associated with the study of metabolic regulation, as well as focusing on improving health care as we move toward e-health.

The focus by IT vendors on the life sciences market, and the tremendous excitement associated with it, began in earnest with the nearly complete sequencing of the human genome, a scientific breakthrough that has opened vast opportunities. For the first time in the history of mankind the book of human life has been opened. Jeff Augen, director of strategy for IBM Life Sciences, points out that whereas the industrial and information eras provided us with tools to make our lives easier and to enable effective communications, the next wave, the biology wave, will allow us to improve health and longevity and create a sustainable environment.

This breakthrough wouldn't have occurred without information technology, which is increasingly becoming an integral part of medical research and of the drug discovery process.

Put another way, biotechnology in the 21st century is characterized by the convergence of the life sciences and information technology, which, it is hoped, will result in discoveries that will improve the quality of health care and the quality of our lives. This is not trivial. We are seeing a true revolution in science – exciting, challenging, rewarding, awe-inspiring.

Market Forces Driving the Convergence of Biotechnology and IT

A number of market forces are driving this revolution, the first of which is the application of new techniques to biological research.

NEW IT TECHNIQUES IN BIOLOGY

Pharmaceutical and biotech companies, as well as academic and government labs, are making increasing use of information technology to handle the large amounts of genomic information needed to accelerate discovery of new targets. Two IT approaches have emerged to address the challenges of generating and dealing with genomic information: one based on automated sequence sample matching and another aimed at computer modeling of biological processes such as protein and cellular interactions. The latter is known as systems biology; it is analogous to the migration that took place in electronic design from using discrete parts to using computer-aided engineering systems. The ultimate migration will be to "in silico" (experimentation by computer simulation). It is this migration to "in silico" research, computer simulation of biological processes, that holds the greatest promise. By using the computer to simulate the experiments and clinical trials that used to take years, we can drastically reduce the costs and increase the efficiency of the drug discovery process. The ultimate goal is to model the disease process at the molecular level. When we are able to accomplish this, we can begin to deliver personalized medicine, delivering the correct drug based on the genetic patterns of the patient. However, we are only on the cusp of this revolution.

We are, however, at a point where the application of information technology to the biological sciences is producing dramatic results. The most obvious of these, as we mentioned earlier, is the Human Genome Project, an international research effort to characterize the genomes of human and selected model organisms through complete mapping and sequencing of their DNA. This project resulted in the coding of the yeast genome in 1998 and the coding of the human genome in 2001.

The reason these breakthroughs are so important is that the human genome not only determines physical characteristics such as the color of our eyes, but also drives the biochemical processes that keep our hearts beating, enable our blood to clot and, most importantly, fight disease. As we begin to understand the structure of the genome, we can begin to understand how it triggers the reactions that fight disease, improving our understanding of how to fight, and ultimately prevent, disease.

Thus the first market force driving the life sciences revolution is the development of new IT techniques in biology. The challenges being faced by the pharmaceutical industry form the second driving force.

CHALLENGES TO THE PHARMACEUTICAL INDUSTRY

The first challenge facing the pharmaceutical industry is really a basic business problem, the time and cost to bring a new drug to market. The average cost of developing a new drug is $802 million, according to the Tufts Center for the Study of Drug Development. It can take over 10 years to develop and get FDA approval for a new drug, and a very small percentage of those approved recoup their development costs.

There are many other statistics: the failure rate of drugs that do appear on the market, the number of those drugs that don't offer a return on the development investment, the number of drugs that complicate rather than heal. One of the most frightening statistics is that nearly 100,000 people a year die taking the wrong medicine. This is the fourth leading cause of death in the U.S., according to a November 1999 Institute of Medicine Report.

The bottom line is that the pharmaceutical industry must change the drug development process. Information technology can help. *Informatics,* which involves taking available data and applying it, comparing and contrasting, enables companies to leverage IT across the drug development process, resulting in shorter cycle times, cost efficiencies, and more products getting to market.

In addition to the cost of drug development, the industry faces other challenges. Drug pipelines are thinning. It is estimated that more than $30 billion worth of drugs will fall off patent protection in the next five years.

Drug profitability is also a challenge. The length of the approval process (in the U.S.) has been marginally reduced, but scrutiny of clinical trials and manufacturing has become more rigorous.. On average, less than 1 percent of all hits identified during drug discovery are commercialized and less than 30% of new drugs recover their R&D costs during their commercial lifetime. And then there is the gap between expenditures and sales. In the past seven years, the top pharmaceutical companies have seen their R&D expenditures more than double, while the number of drugs entering the market have remained about the same. Blockbusters are harder to come by – the industry can no longer rely on blockbuster drugs to cover the expenses and costs of new research, especially as new drugs are targeting smaller populations. When juxtaposed to the declining rate of new chemical entries (NCEs), this suggests a need to improve efficiency in drug discovery that only IT can provide.

There is also the need to increase productivity to maintain growth. Just to keep pace with the annual industry growth rate of 10%, the top 10 global pharmaceutical players will need to launch at least five significant NCEs per year, according to an Andersen Consulting report, "Pharmaceutical and Medical Products Briefing." The industry is not on track to do this.

And then there is the challenge of managing merger activity. Mergers and acquisitions will continue in order to maintain a critical mass of resources. Managing this activity, coordinating resources, integrating data, will be a major challenge, according to IBM's Institute of Business Value. A large percentage of new drugs come from biotechs. However, biotechs lack the marketing infrastructure; thus there will be increasing collaboration between pharmaceuticals and biotechs, which will require additional IT.

While these are significant challenges to the pharmaceutical industry, they represent an opportunity for us, as the use of information technology will be key to meeting these challenges.

In addition to new IT techniques in biology and challenges to the pharmaceutical industry, the last market force driving the life sciences revolution is the move toward personalized medicine and the promotion of wellness.

THE RISE OF PERSONALIZED MEDICINE AND THE PROMOTION OF WELLNESS

Personalized medicine aims to design and deliver treatments tailored to the specific genetic makeup of the patient. If we could associate the genetic makeup of the patient with how specific drugs affect that genetic type, we could take giant steps toward saving lives.

Personalized medicine can reduce the danger of side effects, cross effects from other medications, and the possibility of a wrong drug being received. It can increase the effectiveness of treatments and chances for recovery. It can provide a new focus on diagnostics and preventive medicine for high-risk population groups. We can look to new diagnostic technologies based on genetic makeup analysis.

The overall direction in health care is a shift from treating sickness to promoting wellness. Health care delivery and personalized medicine will require extensive use of information technology. There is an enormous amount of information on patients, information that could be used to move us toward personalized medicine. However, almost all this information is cached in paper files in doctors' offices. Digitizing and correlating this information is a huge task, one that absolutely depends on information technology.

The Opportunity

Why is this revolution in the biological sciences so important to us? The answer is that computational intensity, the ability to manage massive amounts of data and solve computationally intense problems, is critical to success. Consider some of these statistics:

• There are between 32,000 and 40,000 genes in the human genome. Simulating the complex set of events that allows 30,000+ genes to code more than one million proteins is one of the most computationally intensive problems facing researchers.

• The Human Genome Database is approximately three terabytes (one terabyte equals 1024 gigabytes) of data That is equivalent to a stack of paper 17 times as high as Mount Everest.

• If we were to store the data captured by medical imaging electronically, we would be talking about 150 petabytes (one petabyte equals 1024 terabytes) of data annually. The cost in storage alone would run $45 billion, and would require 150,000 systems professionals to manage.

• To simulate the folding of a single protein requires 2×10^{21} floating point operations. This would take a petaflop supercomputer like Blue Gene (capable of a thousand trillion floating point operations per second) a solid month of calculations to code the smallest of proteins, according to Jeffrey Augen.

• The volume of life sciences data in the average biotech company is doubling every six months. Huge amounts of computing resources are required to integrate and analyze this data.

Put in the most basic of terms, the life sciences are going to require huge amounts of computing power. And not just ordinary computing power; we're talking about the power of supercomputers, clusters, and grids.

The Demand for Supercomputing Will Continue to Grow

The advances we've made in supercomputing are astounding. I'm sure you remember Deep Blue, the computer that beat chess grandmaster Gary Kasparov in May 1997. Now consider that Deep Blue essentially had a mental capacity comparable to that of a lizard.

Just three years later, in June 2000, IBM announced that it had built the most powerful ultracomputer in the world. This ultracomputer, called Accelerated Strategic Computing Initiative White, or ASCI White, covers about 12,000 square feet of floor space – an area greater than that of two NBA basketball courts – and weighs 106 tons. The ultracomputer is able to process more operations in one second than a person with a calculator could in 10 million years. ASCI White is smarter than a lizard – almost as smart, in fact, as a mouse.

In December 1999, IBM announced a new supercomputer, christened Blue Gene, that will approach speeds of 1,000 teraflops (1,000 trillion computer operations – or one petaflop) per second, making Blue Gene a thousand times more powerful than the Deep Blue, and about two million times more powerful than today's top desktop PCs.

Supercomputers are key to developments in the life sciences as researchers move toward "in silico" experimentation. One immediate use is to simulate protein folding in an effort to improve the target identification phase of the drug development process. Simulating protein folding is extremely computational intensive, thus driving the need for supercomputers on the scale of the Blue Gene project.

Some people feel this rapid increase in the need for computational power and massive amounts of storage in the biological sciences is comparable to the exponential growth laws we've seen in information technology over the past 30 years. Caroline Kovac, general manager of IBM Life Sciences, said in a Frost & Sullivan interview that if you look at the metrics associated with the increase of biological information, it becomes clear that we are seeing the developments of an exponential scaling law in much the same way as Moore's law represents the exponential increase in transistors per chip.

As a result, we can expect to see an exponential increase in the amount of data and an exponential increase in the computing power and storage required to generate and manage this data. Put another way, the IT market in life sciences is exploding.

Companies Are Meeting These Challenges

Let's look for a moment at how some companies are meeting these challenges. The British Columbia Cancer Agency's Genome Sciences Centre (www.bcgsc.ca) implemented an IBM server and storage solution to maximize scalability for life sciences research. MDS Proteomics (www.mdsproteomics.com), also in Canada, uses Linux-based servers to build supercomputing solutions, enabling advanced analytical capability to determine the function of hundreds of genes and thousands of proteins each month.

Executives at Aventis (www.aventis.com), formed by the merger of Hoechst AG of Germany and Rhone-Poulenc S.A. of France, wanted to create a truly global system to allow scientific resources and results to be shared, regardless of the locations of the individual scientists. Critical to the success of a collaborative environment was an infrastructure that would allow secure access to and optimize querying of heterogeneous data. Working closely with an international Aventis scientific and IT team, IBM Life Sciences Solutions developed a technical and support architecture – including servers, global support capability, and IBM DiscoveryLink middleware – to provide Aventis with the global capabilities to streamline the drug discovery process.

The University of Pennsylvania, in Philadelphia, received grants over several years to develop a revolutionary Electronic Medical Record (EMR) data grid and repository. The challenge was to develop a visionary patient-centric medical record system that could capture the full range of health care files, including high-fidelity patient medical images (CT, MRI, mammograms), records, and clinical history. This meant building a networked system for electronic data capture of patient records; managing and storing huge files for fast retrieval, comparison, and diagnostic review; and ensuring the security and privacy standards required for patient records.

The solution, which includes IBM eServers, DB2 Universal Database, and a GPFS file system, provides secure, scalable, diagnostic electronic patient files in under 90 seconds. Built with open standards, the University of Pennsylvania Grid is a massive distributed computer that delivers computing resources as a utility-like service over a secure Internet connection. Enabling up to thousands of hospitals to store mammograms in digital form, it will also give authorized medical personnel near-instantaneous access to patient records and reduce the need for expensive X-ray films. The system is capable of serving thousands of hospitals. In the future, the University of Pennsylvania will work to extend the grid to additional medical institutions.

IBM's Commitment

This is obviously a huge IT opportunity and IBM is committed to this market. We have over $200 million invested in strategic relationships and development. We continue to form strategic alliances with industry leaders, and collaborate with customers. We have dedicated life sciences teams in sales, development, marketing, consulting, and research, many of whom hold advanced degrees in the life sciences disciplines. We continue to expand our investment in research, especially in the Blue Gene supercomputing project, and in the IBM Computational Biology Center.

IBM Life Sciences is developing solutions to address the IT aspects of drug discovery, drug development, clinical trials, regulatory compliance, and information-based medicine, a system of medical care that supplements traditional opinion-based diagnoses with new insights gleaned through computerized data acquisition, management, and analysis. Using digitized medical images to improve collaboration and analysis between physicians would be an example of information-based medicine.

Our strategy is to leverage our core competencies, and to collaborate to deliver the complete solution. We provide the hardware and infrastructure, the IBM Life Sciences Framework provides the structure for the integration platform. We count on our business partners to provide additional components of the integration platform, as well as the applications and the tools.

Conclusion

The application of technology to the biological sciences will continue to drive scientific discovery. As supercomputers continue to help us analyze the genetic code, high capacity storage will facilitate the storage life sciences data, enabling collaboration between disciplines such as biology and chemistry, and between research teams around the world.

The application of informatics to the drug discovery process will reduce the time to market, and we hope, improve the return on the pharmaceutical investment. We are beginning to make advances both in scientific discovery and in improving the development of new drugs and treatments.

The last revolution – that in health care – lies more in the future. The dream is to simulate "in silico" the role of proteins in causing disease, determining the chemical compounds that will alter that process. Further, by understanding the genetic makeup of the patient, chemicals can be selected that will help rather than harm that particular individual.

This is what makes our work so exciting and rewarding. We are developing the tools and technologies that will enhance the quality of life. Lofty goals, to be sure, but goals definitely worth striving for.

Resources:

• *IBM Life Science Solutions:* www.ibm.com/solutions/lifesciences.

• *National Human Genome Research Institute:* http://www.nhgri.nih.gov.

• IBM Institute of Business Value paper on Pharmaceutical Mergers and Acquisitions at www-3.ibm.com/solutions/lifesciences/pdf/GEE510-1677-00F.pdf .

• *IBM's Blue Gene Project:* www.research.ibm.com/bluegene/press_release.html.

• *ASCI White:* www.llnl.gov/asci/sc00fliers/asci_white_pg1.html.

• *Frost & Sullivan Interview with Dr. Caroline Kovac:* www2.frost.com/prod/news.nsf/LuPages/HCDDT-ibm.

Book excerpt from...

Java Web Services for Experienced Programmers

Welcome to the world of Web services! Over the past several decades, computing has evolved at an unprecedented pace. This progress affects organizations in significant ways, forcing information-technology (IT) managers and developers to adjust rapidly to new computing paradigms. Innovations in programming and hardware have led to more powerful and useful technologies, including object-oriented programming, distributed computing, Internet protocols and Extensible Markup Language (XML). Organizations have learned to leverage the power of their networks and the Internet to gain competitive advantage.

Web-services technology – which represents the next stage in distributed computing – will profoundly affect organizations in 2002 and beyond. Web services encompass a set of related standards that can enable computer applications to communicate and exchange data via the Internet. Although the true impact of Web services is not yet known, many factors – including software vendors' widespread support for underlying standards – indicate that Web services will radically change IT architectures and partner relationships. Companies are already implementing Web services to facilitate a wide variety of business processes, such as application integration and business-to-business transactions.

Since its public introduction in 1995, Java has evolved from a language for building interactive animations for Web pages into a complete suite of software platforms. With its three editions—Java 2 Standard Edition (J2SE), Java 2 Enterprise Edition (J2EE) and Java 2 Micro Edition (J2ME)—developers can use Java to build virtually any type of application. The vast majority of application-server vendors support J2EE for building enterprise applications. J2SE's Swing user-interface libraries and deep support for networking, databases, graphics and multimedia enable developers to build rich desktop applications. J2ME extends the Java platform to small devices, such as cell phones and set-top boxes. With its support for industry-standard protocols, including Web-services protocols, Java is ready for the next generation of interoperable distributed systems.

When the topic of Web services began to appear in magazines, trade papers and industry conferences, we began to anticipate the extraordinary effect Web services would have on the future of interoperable distributed systems. At the 2002 JavaOne conference, there was an entire track with over 140 technical and "birds-of-a-feather" sessions devoted to Web services. For two of this book's authors who attended the conference (PJD and SES), it was clear that Web services would be a primary focus for the future of Java. We hope to provide you with insight into the technologies and standards available to Java developers. These insights, along with practical examples throughout the book, will pre-pare you for developing truly interoperable distributed systems with Java.

Computing Advancements and Web Services

Computer technology is constantly evolving to better serve the needs of programmers and users. Advances in programming and networking have improved software-development and communications capabilities, paving the way for Web services. This section overviews key developments in computing technology that have led to the creation of Web services.

STRUCTURED PROGRAMMING TO OBJECT TECHNOLOGY

One of this book's authors, HMD, remembers the great frustration felt in the 1960s by software-development organizations, especially those creating large-scale projects. During the summers of his undergraduate years, HMD had the privilege of working at a leading com-puter vendor on the teams developing time-sharing, virtual-memory operating systems. It was a great experience for a college student, but, in the summer of 1967, reality set in: The company "decommitted" from commercially producing the particular system that hundreds of people had been working on for several years. Software development is a complicated process, and it was difficult to get software right.

During the 1960s, many large software-development projects encountered severe difficulties. Development efforts typically ran behind schedule, costs often greatly exceeded budgets, and the finished products were unreliable. People began to realize that software development was a far more complex activity than they had imagined. Research intended to address these issues led to *structured programming* – a disciplined approach to creating programs that are clear, demonstrably correct and easy to modify.

As the benefits of structured programming and of the related disciplines of structured systems analysis and design were realized in the 1970s, improved software technology did begin to appear. However, it was not until *object-oriented programming* became widely used in the 1980s and 1990s that software developers finally felt they had the necessary tools to improve the software-development process dramatically.

What are objects, and why are they special? Object technology is a packaging scheme that enables programmers to create meaningful software units. These units can be large and are focused on particular applications areas. There are date objects, time objects, paycheck objects, invoice objects, audio objects, video objects, file objects, record objects and so on. In fact, almost any noun can be reasonably represented as a software object. Objects have *properties* (i.e., *attributes*, such as color, size and weight) and perform *actions* (i.e., *behaviors*, such as moving, sleeping and drawing). *Classes* represent groups of related objects. A class is to an object as a blueprint is to a house. A class specifies the general format of its objects; the properties and actions available to an object depend on its class.

We live in a world of objects. Just look around you – there are cars, planes, people, animals, buildings, traffic lights, elevators and so on. Before object-oriented languages appeared, procedural programming languages (such as Fortran, Pascal, Basic and C) focused on actions (verbs) rather than objects (nouns). This paradigm shift made program writing a bit awkward. However, with the advent of popular object-oriented languages, such as C++, Java, C# and Visual Basic .NET, programmers can write code more naturally in an object-oriented manner that reflects the way in which they perceive the world. This process has resulted in significant productivity gains.

One of the key problems with procedural programming is that the software units created do not mirror real-world entities effectively and therefore are not particularly reusable. Programmers often write software for one project then rewrite similar software for other projects. This wastes time and money as developers repeatedly "reinvent the wheel." By contrast, object-oriented programming allows for code to be organized and encapsulated as classes, which facilitates the reuse of software components. Developers can group classes into *class libraries*, which they can then make available to developers working on other projects.

Web services extend the object-oriented paradigm, in that objects can represent Web services. Every Web service is, potentially, a self-contained object that can be reused by, or incorporated into, other applications. In this manner, the technology encourages a modular approach to programming, transforming a network such as the Internet into an enormous library of programmatic components available to developers. This can greatly reduce the effort required to implement certain kinds of systems—especially compared with the effort involved in reinventing these capabilities for every new project. ✦

WebSphere DEVELOPER'S JOURNAL | **Coming Next Month...**

WEB SERVICES: A PRACTICAL APPROACH...
How to add value to your enterprise resources using a Web services architecture

CREATING MESSAGE-BASED WEB SERVICES WITH WEBSPHERE STUDIO APPLICATION DEVELOPER, PART 2
Take the steps necessary to introduce asynchrony into the use of SOAP message-based Web services

BUILDING DB2-BASED WEB SERVICES USING WEBSPHERE, PART 2
Web service-enable your DB2 database and leverage your investment in the database schema

PERSONALIZE YOUR WEB SITE WITH WEBSPHERE PERSONALIZATION
Provide a personal touch and a superlative user experience by extending the intelligence of your Web application

EVENT-DRIVEN ENTERPRISE JAVABEANS
Using an EJB as a user thread, you can develop event-based systems in any EJB application server

Web Services for Faster Deployment?

Applying the lessons of the past

BY NICOLAS **JABBOUR**

Faster deployment. Who would say no? It's the dream of every project manager, business line manager, and CIO to get software out in the shortest time, at the lowest cost. However, it's an ideal that is rarely met.

Project members each look at "faster deployment" differently, using the best tools to help them accelerate their own tasks and responsibilities, from monitoring to debugging to testing. Can we conclude that by using the best-of-breed enabling tools at each step of the project, the whole project will deploy faster? Unfortunately, no.

But there are two factors that can aid you on the road to "faster-deployment nirvana" – project management and reusability.

Let's think about project management. Those of us involved in this know that minimizing interdependencies among the teams will yield a better handle on the project – and therefore better results. Each group focuses on their area of expertise and commits on their deliverable, shielded from the deliverables of other teams. Henry Ford used this approach of "subdivision of labor" in the early 20th century when he revolutionized manufacturing. Ford's declaration "I will build a car for the great multitude" was realized; his project management style helped him gain huge productivity benefits and essentially turned the car into an accessible commodity.

Let's look at reusability. In the electronics industry, progress has been dramatic over the past 20 years. There are more and more products for cheaper and cheaper prices! For example, take the modem card, which is now designed from a "central catalog" of standardized, predictable, and documented components – a completely revitalized process that yields great value to the manufacturer and consumer.

Where are we in the software industry? Not that far along. Although we have well-structured project management phases – from discovery to design to development to deployment – the process inevitably contains major pitfalls. There is a double gap between "user and analyst" and "analyst and developer," leading to business requirements getting further from the end result. There is no central place to publish the available components for further reuse, and there is a high dependency between front-end designers and back-end integrators, which can make interactions amongst the teams challenging, and can even delay the project. Additionally, object-oriented languages and components certainly didn't achieve the reusability dream as we first had hoped.
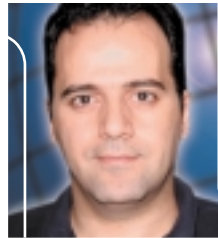
Since components were often technology-related, their inherent usage constraints limited reusability and therefore reduced adoption.

However, with Web services we have many of the factors needed for success. The emergence of a service-oriented architecture (SOA), combined with standards such as Web services, can cut development costs, yielding projects that are deployed faster and cheaper. This process involves creating an enterprise repository of services, published in an industry-accepted standard, Web services. The repository holds all the business services the enterprise wants to make available and share across divisions or business partners. The adoption of Web services as the standard implies that the SOA is implemented using UDDI for the repository and WSDL for the description of the service interface or signature. The immediate advantage of the Web services–based SOA is that it holds descriptions of any service, regardless of its implementation, and can interact with any service consumer, regardless of its technology.

With this approach, project management now achieves the benefits of subdivision of labor. During design, the analyst specifies the desired functionality along with interfaces that are validated by the users. These interfaces feed the UDDI so you can immediately declare functions and use generic "stub" implementations to make them instantly accessible. This becomes the "contract" between those who implement the business services and those applications that consume them. Suddenly team dependencies are eliminated – back-end developers implement the services using their desired approach and technology, while front-end developers begin their work without any delays. Also consider how this approach enables business sponsors to review functionality sooner.

With this approach, it's also easy to see how reusability becomes a reality. All services are defined and declared in a central, technology-agnostic place. Deployed services are stored in the UDDI repository and the WSDL holds the documentation and services signatures. New projects can start and use these existing assets. Suddenly you have an open, flexible infrastructure that grows with your organization's business demands.

Perhaps by reusing the lessons learned by others before us, and applying new advances such as Web services combined with a services-based repository, the "faster-deployment nirvana" is attainable for software too.

ABOUT THE AUTHOR... Nicolas Jabbour, CTO of Prolifics, is responsible for architecting WebSphere solutions for global, cross-industry clients. He leads the Prolifics WebSphere Consulting Division, a staff of specialized consultants retained by IBM to solve the toughest of their customers' challenges and deliver training, mentoring, and development services. Nicolas is a well-known speaker on the topics of WebSphere, business integration, Web services, and portals.

E-MAIL
njabb@prolifics.com

TRILOG GROUP

WWW.FLOWBUILDER.COM/XSP