WebSphere ADVISOR MAGAZINE

MARCH 2001

APPLICATION INTEGRATION

Insider Interview: Plan for Web Services

Two companies discuss how you can prepare for Web Services.

By Christa L. Coleman, Senior Managing Editor

Web services are self-describing, modular applications that work seamlessly with other Web services to fulfill a specific task or a set of tasks. In essence, they enable just-in-time application integration. The idea behind Web services is that you'll have access components that you can not only publish, but find and invoke across the Web without prior knowledge of the application or its programming requirements.

I recently spoke to representatives from two IBM Business Partners, Prolifics and Rational Software. Both companies are working to make everyday use of Web services a reality. As a result of their work with Web services, they offer interesting perspectives on how you might use Web services, how Web services might evolve, and most importantly, how you can prepare.

Nicolas Jabbour, Prolifics

Advisor: How can companies be plan for Web services?

Jabbour: The best way to build these systems is to use a component-based architecture, such as WebSphere and EJB, because it maps immediately to this business need. If you have a system that is procedural, or in some other way not component-based, it will be difficult to open it in a Web services way, although not impossible.

Because the components talk to each other in a standard way, you can give them an XML interface that maps to the Service. Think of it this way: Web services gives to the end client what component-based development gives to the developer.

Advisor: How do you describe Web services to your customers?

Jabbour: The concept of Web services isn't new. It's what everybody has wanted for a long time, we just didn't know what to call it or how to do it. Plus, the technology, to some extent, wasn't mature enough to enable this kind of application interaction. I would say that Web services is the business

ADVISOR THOUGHT

Perpetual optimism is a force-multiplier.

-- Colin Powell, American politician

Nicolas Jabbour

Nicolas Jabbour, VP of e-Solutions for Prolifics, manages a staff of thirty specialized WebSphere consultants who support AIM Services, a division of IBM which delivers specialized WebSphere training, development, and deployment services. He also guides the creation of enterprise-level, mission-critical WebSphere applications. Prolifics is provider of solutions, services, and technologies that transform businesses into e-businesses.

Jeffrey Hammond

Jeffrey Hammond is senior product manager for Rational Software. Rational Software Corporation helps organizations develop and deploy software for e-business, e-infrastructure, and e-devices through a combination of tools, services and software engineering best practices.

consequence of the technology maturing from structured to object-oriented to component-based.

I've been working with IBM in France on a large project for the ministry for social security. For this project, it wasn't just about choosing WebSphere for the middleware, it was about how to best take advantage of WebSphere in a Web services context--for this project, it was a real advantage.

This project is a great example of how Web services can be used. If you are the French social security ministry, you can expose the medicines you sponsor, along with their prices. Any doctor using any system can go see what the generic and brand name medicines to recommend. If you expose your internal systems via Web services, any doctor with any standard interface--in this case it's based on XML can get this information from you without directly interacting with your internal systems.

Another example: Say a French citizen has his medical records stored and formatted the way the French government does it. If he comes to the U.S. and needs health care, the hospital system here can check his records by asking the service that's published by the French Ministry of Health.

Real life examples of how systems worldwide can communicate with each other in a smart, loosely coupled way are what get people to understand this technology makes sense.

Advisor: What is the biggest misconception about Web services in the marketplace?

Jabbour: That Web services is about technology. It isn't. It isn't about asking component A to talk to component B. It's about building systems in a smart way. It's about companies that don't have the resources to build financial and HR systems, and don't have the resources to buy SAP and PeopleSoft and make them work together. They can subscribe to providers who let them, when they have a new employee, go to the HR system they subscribe to and, with even a homegrown application, create a new record for this employee. They then go to the payroll application they subscribe to and have it generate his payroll at the end of the month. These two systems are completely separate.

Advisor: How will this affect the role of the developer?

Jabbour: The role of the programmer will be to develop modular systems, and he will go and look for products like WebSphere. He will look for best practices for developing and linking EJBs.

The business person will be focused on finding services that satisfy business needs. The services will differentiate themselves on functionality, price, etc.

Advisor: How do you think Web services will evolve?

Jabbour: To start out, Web services will work best in vertical markets, for example the medical world. In a limited environment like this, you can make a population across boundaries, for example, the U.S. and France, agree on a common language, a common way to exchange data, etc. Based on this, you can create services in each country that can work with each other to exchange files, information, etc.

Advisor: Is the market realistic about what Web services can offer?

Jabbour: Web services exists as a prototype today. However, the concept of Web Service is so nice, and so pure, it shouldn't fail. Prolifics has been in business 22 years, and we have seen the same concepts before, it's just that the technology terms are

different. We used to say "use CICS this way," or "use this OLTP system that way," but the point was that there was a decoupling between the caller and the receiver. We used to format the message, not in XML, but in EDIFACT. So the technology names changed, but the concepts are the same. The concept is very simple. This is its beauty.

But, you have to start with simple things. Instead of integrating B2B systems, you have to start with something like the medical example I mentioned. Or, you can have less complex systems deployed to a larger business community. To start out with, the expectations have to be in line with that.

The other issue with a large community is the number of Web services available for a particular business need. Who is going to map them all to each other? When you go looking for a payroll system and you get 2,000 of them, how do you filter, how do you figure out which one best fits your needs? So, the next step is to figure out to deploy this massively.

Jeffrey Hammond, Rational Software

Advisor: How is Rational approaching Web services?

Hammond: One of the things we're focusing on is better integration with the WebSphere runtime, so we have a lot of development tools that are complimentary to IBM's development tools. We're working closely with IBM to establish how those tools work together to focus on getting content out to the WebSphere runtime as quickly as possible. For example, our tool, Rational Rose, lets developers visually design these types of systems.

In the Web services world, Rational Rose lets you work with the XML document type definitions (DTDs) that are a part of Web services. So, you can visualize the UDDI DTD in Rational Rose and use it to understand how it's organizationally structured. It's very helpful when you're trying to understand how Web services are put together as you move forward to design your own Web services for the platform.

Advisor: At what point did you start working on Rose to update it to accommodate Web services development?

Hammond: We started working on it from a generic standpoint. The nice thing about Web services and the UDDI initiative is that it's built on standards-based technology, namely XML. So, we didn't have to start completely from scratch--there were existing pieces we were able to use to assemble these targeted parts of the solution. A key strength of the Web services architecture is that you aren't starting with a completely proprietary technology you have to assimilate and understand.

Advisor: What does Rational consider what it calls the "full e-development lifecycle" to be?

Hammond: You're really seeing the convergence of a couple worlds. You can't develop production-level enterprise Web applications without using Java and J2EE and integrating with existing solutions. That's one area. You also have content developers and people who are writing HTML, who might also be using templates, and it's traditionally a more content-oriented world. We're starting to see these two worlds coming together into a continuum, so you need a business development process that integrates the work of the code developers and the content developers in a seamless manner.

Advisor: What's your assessment of Web services?

Hammond: It's a logical and reasonable extension of standards

that have been put into place. I think people are having a hard time understanding what UDDI is, how we're going to be shipping all this information back and forth on the Web, in real time, etc. But, the reality is that we've been doing this for a while, it's just that it's been in relation to presentation and customer-based information. We're just kicking it up a notch, formalizing it with XML, and making it conform with a standard. So, it's really an evolutionary step, not a revolutionary step. And, it's one that makes sense.

MIW0103.COLEC84

ADVISOR THE SITE FOR EXPERT ADVICE™

Welcome to Advisor Academy®

----COMMUNITY----