Notes.net

Iris Today

John Banks-Binici on Domino and WebSphere integration

by Amy E. Smith and Dick McCarrick

Level: Advanced
Works with: Domino 5.0
Updated: 04/03/2000

[Editor's note: John would like to know what you think about Domino and WebSphere integration. After reading this interview, visit the ***Developer Spotlight Forum*** *to talk to John.]*

John Banks-Binici is leading the effort at Iris to integrate the Lotus Domino and IBM WebSphere server products. In this interview, he talks about ...

What is WebSphere?
WebSphere is a Web application server developed by IBM. It offers a standards-based Java server run-time environment and a set of Java development tools.

Currently, there are three different server editions in the WebSphere product family. There's the Standard Edition, which comes installed, at no cost, with Domino R5.

Next there's Advanced Edition, which builds on Standard Edition by having an Enterprise JavaBeans programming environment. This environment allows you to write portable Java component-based applications with easy-to-use builder tools. With these tools, you can build applications that take advantage of complex server features, like two-phase commit transactions, which are critical to many applications, especially the new breed of customer-to-business and business-to-business applications. Advanced Edition also includes a Java Object Request Broker [ORB] for building networked Java objects.

Finally, the Enterprise Edition adds the Component Broker C++ ORB, which allows you to write and deploy C++ based server objects as well.

How will customers typically use WebSphere?
Think of it as a Java server platform. It provides a set of services for accomplishing the usual things that server developers need to do in a portable way. To write truly portable apps, you need to rely on a consistent intersection of services that are available on all the platforms you want to run on. That means you have to avoid platform dependencies beyond those abstracted by the the Java Virtual Machine. For example, you want your Java programs to access the directory via standard APIs. The Java Naming and

Directory Interface (JNDI) gives you that ability. So customers will use WebSphere when they want to write portable, Java-based applications. They'll use it to leverage the Enterprise JavaBeans (EJB) programming model, and they'll use it when they want to write applications that require highly scalable transactional capabilities.

What's the nature of your involvement in all of this?
Several months ago, there were a lot of rumors going around among our business partners that there was an impending train wreck between the Domino and WebSphere products. Much of this was caused by confusing signals being sent out by Lotus and IBM. Those of us who understood the issues felt that we needed to make a concerted effort to educate people, both inside and outside the company, about how these products relate to one another. We needed to clearly delineate the strengths of each product, and take it one step further and capitalize on the enormous opportunities that existed by combining their strengths. I was asked to be the technical lead representing Iris Associates. I understand the value that these products bring to the table and can see their strengths and their weaknesses. My job as an architect is to get the word out about the strengths and address the weaknesses through joint development with IBM.

Why integrate Domino and WebSphere at all?
The most important reason is that customers are demanding it. People want all of the capabilities that these two products have to offer, and they want us to iron out the issues to allow them to work together. For example, if you've already got Domino deployed and you're using Domino Directory, you want to be able to add the capabilities that WebSphere has to offer and leverage your existing directory, security, and administrative infrastructure.

Does WebSphere compete with Domino? For example, are there plans to migrate some Domino customers to WebSphere?
They don't compete. What confuses people is the lack of information about how these products work together and complement each other. In the past, there has been so little information available that even internal folks have been confused. In fact, these products have tremendous synergy. You can use the combined products to deliver applications that leverage the unique aspects of collaborative and transactional applications.

Regarding migration, again, that idea is really a symptom of the confusion I'm talking about. People tend to assume that just because we could make this all one product and we're not, it means we're going to compete instead.

Even though there is tremendous synergy between these products, it makes the most sense to let them remain two distinct products because each pursues some market segments that the other doesn't.

It's important to understand that the goal is still to sell the products as natural complements to one another and as part of a larger family. Both products are just plug-ins to the customer's HTTP server; we share the same HTTP stack. Today that HTTP stack can belong to either Domino or Microsoft IIS.

We have strategic plans to expand this capability to support other HTTP stack configurations -- Apache, for example. Ultimately, the goal is to make it transparent which product actually delivered the bits you needed. They'll share common components so that if you're already using one product, you'll be able to easily buy and add the other.

How will customers decide which product is right for them?
It really depends on what you're trying to build. If you're building Web applications that focus on collaboration, Domino is the right choice. If you're building a Web site that needs to handle a lot of transactions or requires the

use of Enterprise Java components, then WebSphere is the right choice. If you need to build a site that needs both of these capabilities, Domino and WebSphere work together over the same HTTP stack to deliver the best of both worlds.

For example, one customer has built an application that uses a Domino workflow infrastructure to implement content management for their Web site catalog. Domino-based workflow and collaboration software is used to manage the content generation and content approval process for catalog entries and prices. The highly scalable transaction services of WebSphere are used in tandem to implement the actual catalog site. Domino messaging is used to send confirmation information to the customer and to maintain the dialog with the customer.

So you can imagine the flow of information. Here's an over-simplified example: A buyer for a major catalog site has finally found some special blue widgets that they were told by marketing were in very high demand.

The buyer forwards the price and photograph to their marketing contact. The marketing person agrees that this is good news and then routes the workflow to the company's Web page designers who build an edgy new catalog page for the item.

The designer routes the page back to the marketing person who initially approves it and therefore routes it to his manager for final approval.

The marketing manager approves the catalog page with a click of a button, and thus automatically posts the page to the company's catalog site. So far it's all been Domino. Next, a customer visits the site and buys the product.

Transactional business logic running in WebSphere automatically adjusts the inventory count, debits the credit card, and routes the order to shipping for fulfilment. It also uses Domino to fire off an e-mail message to the customer confirming availability.

The shipping clerk receives a workflow message that the widget should be shipped to such-and-such address. At the end of the day, the widget goes out on time, and with the click of a button, the shipping clerk ends the transaction.

If the customer has questions about the status of the order, they generate an e-mail reply to the confirmation or visit the site. As this example shows, the two products are meant to work together.

Will many customers use both Domino and WebSphere at their sites?
At this year's Lotusphere, we conducted a Birds of a Feather session that started at 7 AM. It was standing room only, and one-third of the attendees had to participate from the hallway. Many of those customers were in the process of building meaningful applications, and they were there to get their questions answered. So the answer is yes. There are a lot of people that are excited about what's possible when these two products are used together.

In general terms, how will the integration be implemented? What areas will be integrated?
For starters, we're working to define a single sign-on capability that spans these two products. So we're going to make it explicit how someone should write an application that uses services from either Domino or WebSphere, while only requiring a single user login. That means you'll be able to initiate a transaction from WebSphere and have the backend execute some aspects of it on Domino under the same login credentials. The opposite will also be possible; you'll be able to start a transaction on Domino and have some aspects of it execute in WebSphere, again under the same identity. The goal is to provide a seamless flow of information across these products so that

the boundary defined by which bits get shipped in which box becomes unimportant for the purposes of building applications.

We're starting at the foundation for building highly integrated applications, beginning with a common directory and security based on what's already available in Domino. We're going to move up from there and work on integrating administration. Along with that, we're merging our programming models via several initiatives. The single sign-on initiative is one example; another is that WebSphere will be integrating and shipping LotusScript (Lotus's BASIC scripting language) support. This will allow users to write LotusScript within WebSphere's Java Server Pages (JSPs). That's going to allow a whole class of developers who are familiar with the BASIC language (including Microsoft's Visual Basic) to leverage their skills in writing WebSphere applications. Domino will also be leveraging JSP technology.

How will this benefit Domino users? WebSphere users?
The benefits really accrue to the developers and the administrators who will be able to craft and deploy a whole new breed of applications that leverage Enterprise Java, the Domino infrastructure, and its very large installed base. Of course, the ultimate benefits will accrue to the users who will be empowered via the new breed of application that is possible when you combine the strengths of collaborative and transactional functionality. It's precisely the sort of environment you need to build an e-business.

How will Domino Designer coexist with WebSphere Studio?
Domino Designer is, and always will be, the way to write applications that reach Notes clients in addition to Web clients. We have a number of initiatives to make the two products share common technology. For example, the initiatives I've already mentioned are common LotusScript and JSP capabilities. We're also working on common ways to manage the development process, for example, debugging features and capabilities across the two products.

Can you speak about timeframes?
We're trying to complete functionality that helps customers do single sign-on as soon as possible, and WebSphere intends to ship LotusScript capabilities in their 4.0 release. More work will follow in both products, but I can't comment on those plans yet.

It sounds like you've made progress on understanding the issues for integration, and now you just need to get it scheduled before you can talk about it in more detail.
That's largely true. The process is well underway, but we're nowhere near done because this is really a multi-release effort. One of the benefits of all of this work is that it's greatly improving our technical-level relations with IBM. I'm glad to be associated with the project.

*[Editor's note: John would like to know what you think about Domino and WebSphere integration. After reading this interview, visit the **Developer Spotlight Forum** to talk to John.]*

ABOUT JOHN
John Banks-Binici is a software architect at Iris, and has worked on much of the design and implementation for the NTI networking layer in Notes. He lead the server mail and directory teams for R5. He's currently working on WebSphere integration, and he also plays guitar in Iris's rock band the "Iris Notes."