Title: Composite applications for today's Eclipse and Lotus Expeditor developers.

Abstract: This podcast is targeted to Eclipse™ application developers who can benefit from integrating Eclipse platform and Lotus® Expeditor software-based applications with existing IBM Lotus Notes® and Domino® software-based applications. We'll discuss how much Notes® and Domino knowledge is needed to do this integration. The podcast also discusses use cases and then covers the basic steps involved in building and deploying a composite application in Lotus Notes 8 software. (The Eclipse foundation is an open source community committed to implementation of a universal development platform.)

Interviewer: Mark Jourdain

Presenter: Bob Balfe

Length: 13 Minutes 59Secs

Size: 13.1 Meg

Transcription:

JOURDAIN: Hello. Welcome to this IBM podcast, Composite Applications for Today's Eclipse and Lotus Expeditor Developers. I'm Mark Jourdain from IBM Software Group, Lotus Software, and I'm joined today by Bob Balfe, one of the architects for composite applications and Lotus Notes [software].

BALFE: Hi, Mark, glad to be here.

JOURDAIN: Bob, let me ask you this question first. This webcast covers composite applications. Does this mean it covers "mashups" too? Are mashups and composite applications the same? And also, how do composite applications relate to SOA?

BALFE: Well, it really depends on how you define a mashup and how you define a composite application. Both mashups and composite apps can leverage data from different applications and systems.

Recently, mashups have been more associated with Web technologies -- like RSS [Really Simple Syndication] feeds or ATOM feeds -- or integrating Web-based components. Composite applications, however, aggregate presentation-oriented components side-by-side with each component having its own defined business logic and ideally a standardized interface for wiring and interaction with other components.

Many call composite applications, you know, the business mashups, as they allow IT and authorized users to make use of clients like Lotus Notes to provide a single customized view that includes different applications and systems from servers and the desktop. They provide the right information for that user and help to improve the user responsiveness.

As far as SOA, composite applications can be called the human interface to Service Oriented Architecture [SOA], as the components provide independent services and can be loosely coupled -- wired -- for interaction via their interfaces.

JOURDAIN: So Bob, I've heard a lot about composite applications. You know, you just gave us some more information about it. If I'm already building Eclipse-based applications, why would I consider composite applications in Lotus Notes 8?

BALFE: Well, the new architecture of Lotus Notes 8 incorporates a lot of

Lotus Expeditor [software], which is built on top of the Eclipse Rich Client Platform. Notes 8 client software provides a composite application framework. It's a selection of runtime environments and a readily-accessible deployment and administration model from a Lotus Domino infrastructure.

For the first time, Eclipse developers can basically supply capabilities and solutions to the large installed base of Lotus Notes users. From a business perspective, there are additional benefits.

One, composite apps [applications] and Lotus Notes can help end-users be more effective as they work in an application that streamlines their interaction with multiple systems and applications as a single customized interface.

And two, developers and the IT organizations can have long-term benefits of quicker application delivery and potentially reduced cost with components that can be reused in multiple composite applications.

Also, as a business partner or system integrator you can provide components that expose your application domain to Notes users. If you want some more information on that, you should really listen to the business benefit podcast of this series.

JOURDAIN: Great. So Bob, are composite applications the only solution that supports integration with Notes or Domino?

BALFE: No, as with all prior releases of Notes and Domino there are programmatic back-end capabilities provided through technologies like the remote Java APIs, the COM, C APIs and Web services. You can get more

information on these capabilities by reviewing the Domino Designer help and C API toolkit documentation also available online, www.ibm.com/developerworks/lotus/documentation.

JOURDAIN: So, Bob, what is a simple composite application use case that I can make use of to advocate the value of composite applications with Notes?

BALFE: Well, the Notes 8 mail and Notes 8 calendar applications are already composite applications and can easily be extended with additional wireable components. I'll give you an example.

You can add a component to your mail that serves as a document management system. The Notes 8 mail and calendar components can also be used in other applications to show content from either the current user's mail, calendar or from a shared Notes application, such as a mail-in database for instance.

JOURDAIN: Great. Now, are there any other areas, you know, that you'd recommend as good candidates for a component?

BALFE: Well, you should consider your enterprise systems that provide valuable capability such as business intelligence applications, enterprise data sources, document management systems or even host applications for components.

JOURDAIN: Bob, I think a question that's probably come into a lot of people's mind is, how much Notes and Domino knowledge do I need to have to integrate Notes applications with existing Eclipse Rich Client Platform applications? You know, I mean, if I already know WebSphere® Portal, are there

other things I need to know?

BALFE: Well, similar to composite applications on portal [IBM WebSphere Portal software], what you really need to know are what properties can be published from a Notes application, and what wireable actions or a Notes application can provide, plus their data types and name spaces.

For instance, if the properties and actions you require are not already available, a Notes developer could extend the application to provide those and then provide the business logic to support the properties and actions.

If you are already building portlets, for instance, for WebSphere Portal, you can use the [Lotus] Expeditor toolkit to export it as a feature and its plugins. Or the Notes client can then consume those Eclipse features, and the portlets would be rendered in the portlet viewer runtime to display the portlet.

JOURDAIN: So Bob, I know the Eclipse programming model. What are the main things I need to know to build one of these wireable components that you mentioned?

BALFE: Well, you should have Notes 8 already installed and set up on a machine that has your Eclipse IDE [Integrated Development Environment]. If you don't have Notes already, you can download trial versions at www.ibm.com/developerworks/lotus/downloads.

Eclipse-based composite application components in Notes 8 extend a ViewPart [from Eclipse]. So you just simply write your views as you normally would. One of the primary differences is since it can be rendered in multiple applications, you

need to have the allowMultiple="true" flag in your plugin XML to be used as a component within a composite app [application].

You can write the user interface as a portlet, AWT, SWT or JFace user interface. You define a WSDL -- which is a Web Services Definition Language file [Editor's correction: Web Services Description Language] -- which defines your properties and actions for the component so it is wireable to other components.

And then you just simply include the property broker plugin as a dependency and then tie the wire into the property broker extension point. And then you create a new feature and an update site for your project or your feature, just as if you would to deploy any other Eclipse-based application.

Many of these steps can be streamlined using Lotus Expeditor toolkit. There are wizards and export options available -- conversion tools for portlets, for instance. And the toolkit also provides a debug and launch profile for Lotus Notes to configure Notes 8 and use Notes 8 in your test environment for debugging.

JOURDAIN: OK. So after I have this wireable component, how do I actually get a composite application? How do I create a composite application from these components?

BALFE: Well, for an NSF-based composite application, you create a new NSF application using the composite application template option supplied with Notes and Domino. [Editor's note: NSF is Notes Storage Format.] This option is available from the Notes client using "File > Application > New," then selecting the template labeled "Blank Composite Application."

After you open the new application, you can then use the Composite Application Editor feature to edit it. This is an optional feature for Notes 8 and can be enabled for some users and not others.

What the Composite Application Editor feature does is it provides a graphical interface to let you define the layout of the components and define the wiring between the properties and actions of the components on the screen.

For a WebSphere Portal-based hosted composite application, you use the portal templates option from the Open list of the Notes client. (This option only shows up if you've actually configured your Notes client to connect to a portal server.) You can then go ahead and create a new application based on the selected template. You can use the Composite Application Editor, just like in the previous example, to edit the application, lay it out and wire the components together.

To edit an existing composite application, just simply open it and then you open up the Composite Application Editor from the Notes client menu. You'll be able to edit the application as long as you have the proper permissions to do so.

JOURDAIN: Now, once I have this composite application, how do I make it available or deploy it for the end-users?

BALFE: Well, it's important to keep in mind that Notes clients interact with a Domino server for mail, calendar and custom business applications. A composite application is no different.

If you've created an NSF-based composite application, you provide your Domino administrator either with a copy or the replica of your NSF-based composite app

[application] and you place it on the Domino server.

You provide the Domino administrator with any additional NSF applications that are needed for your composite app, and you just simply put it out on the server -- including things like the NSF-based Eclipse update site, if that's how you're going to deploy your plugins and features, for instance.

End-users can then use the Notes client menu to access the NSF-based composite application. Or the Domino administrator can also use the Domino directory policies to add NSF-based composite apps and other NSF applications to the workspace of the Notes client.

Now, if you went ahead and created the composite application on portal, end-users would use the portal applications options from the Open list menu in their Notes client. And then they'd simply select the application they want to open right from the portal catalog.

JOURDAIN: So, another question: what are the best options for components if I want to run these on both WebSphere Portal and in Lotus Notes?

BALFE: Well, Lotus Notes 8 comes with the same portlet container that ships with the Lotus Expeditor. What this means is now if you deploy your portlets as Eclipse features and plugins, the Lotus Notes 8 client can render those portlets locally within that container.

So for instance, if you have portlets already for WebSphere Portal, you can transform the existing portlet Web application archive -- or the WAR file -- into a portlet Web application bundle through the use of the Web utility which ships with

the Lotus Expeditor toolkit.

JOURDAIN: So, Bob, where are examples and other information I can learn from? Are there any preassembled samples or components that I can download and use?

BALFE: Yes, absolutely. There are a few resources you should find especially helpful. One, there's overview information. It's available on the Lotus Notes Composite Applications page which is at this URL, www.ibm.com/software/lotus/products/notes/compositeapplications.html ["compositeapplications"] all one word -- dot-html.

There's information included in the Lotus Notes Composite Applications page off of [IBM] developerWorks™. There's a table of contents linking to multiple resources. You can see the program availability link from this page.
And in Q1, you should look for a new wiki, which is planned to help developers with the process and procedures for composite application and component development and deployment. The URL for that developerWorks page is ibm.com/developerworks/lotus/composite-apps, A-P-P-S. You can also find information specific to NSF components in the Domino Designer help. And administration of composite applications in the Domino admin help.

[Editor's note: The wiki is now available at the following site: http://www.ibm.com/developerworks/wikis/display/appdev/Home]

JOURDAIN: So Bob, I think I heard you mention other podcasts earlier. Are there other podcasts that the listener today would find interesting?

BALFE: Absolutely. I would also recommend Podcast 1 of this series, The Business Value of Composite Applications. It provides more information that helps explain to CIOs and IT managers on how composite applications can help enable business value and discusses IBM's strategy for supporting composite applications.

We've also created another podcast called Composite Application for Lotus Notes and Domino Developers. Developers in your organization with Notes and Domino skills will be interested in listening to this to learn how they can leverage those skills and extend them with the new capabilities of the Lotus Notes 8 software.

JOURDAIN: Thanks Bob. And thanks to those of you who have listened to today's podcast.

For more information, please visit the Lotus Notes and Domino product page, ibm.com/lotus/notesandomino, and follow the links to other resources we've discussed.
[END OF SEGMENT]