



Announcer You're listening to the Sun Microsystems Podcast Network. Welcome to another edition of *Innovating@Sun* with your host Hal Stern. Today's topic: *JavaFX*. And now here's Hal Stern.

Hal:

Hello and welcome to another edition of *Innovating@Sun*. I'm your host Hal Stern, vice president of Global Systems Engineering. And today I have Bob Brewin, the tallest chief technology officer at Sun Microsystems. So, tall guy, welcome to the show.

Bob:

Hey, how are you doing? Aren't you one of the short CTOs?

Hal:

Well, I try to make up on the weight end of the requirement rather than the height end. So, you know, volume is always preserved.

Bob:

Well, you don't have to worry about the "taller they are, harder they fall" problem.

Hal:

So, Bob, you've been running the software platform side, everything above Solaris, now for a little over a year. And *JavaFX*, I know, is something that you've taken a personal interest in shepherding during that time and certainly a tremendous amount of excitement around it. Give us the overview.

Bob:

So I've been here for about a year, so right after Java I last year, I took over this new position and obviously there's been a whole lot of things going on in software. I think one of the most exciting ones of late is *JavaFX*. Now *JavaFX* is interesting because as Rich Green used to quote things, it's a dessert topping and a floor wax in many ways because what it is, is it's a product family from Sun, a brand new one, specifically designed around delivering some rather compelling and interesting software solutions for folks. There's things like *JavaFX Mobile* and *JavaFX TV*. There's going to be a whole set of runtime solutions where if you think about it in the sense of Java existing on top of Solaris, gives you a complete runtime solution for building on top of Java. Apply that to things like the mobile space and apply that to the TV space. Now, from my background, before I joined software group as a CTO, I used to work on the developer side of the side, and so obviously, you know, the *JavaFX Mobile* piece is largely a deployment story. On the development side, I think we have some rather interesting and exciting stuff happening in *JavaFX Script*, which is so how do developers build sort of compelling and rich user interfaces, leveraging Java for a wide spectrum of devices? In fact, you know, one of the promises behind *JavaFX* in many ways is we might actually start achieving this "right one for an anywhere promise" that we tried to deliver 12 years ago.

Hal:

So you talked about *JavaFX* really sitting in a product family, and I think the historical view of Java was it was

something that allowed you to write network clients for desktops. And then you had Java ME that would allow you to write a different set of network clients for phones. And really, I think we've seen this explosion of different kinds of network clients with different profiles, with different varieties of graphics capability. Where does Java fit in with unifying that world?

Bob:

Well, so here's the interesting thing – and I'm going to actually point people to my blog which is on blogs.sun.com/brewin. There's an article there which entitled Back to the Future. Now Back to the Future was something that I used as sort of the backdrop for the keynote presentation at Java I. The interesting thing about that particular presentation is it really demonstrated visually as well as verbally sort of this cycle that we go in sort of waves of innovation against the Internet. And part of that is this notion of clients and clients that are connected to the network. And so one of the interesting things is we take a look at all of these things moving forward, is we've really got to a point now where we're moving from things that are connected to the Internet in the sense of, let's say, rich Internet applications but largely on desktops and webtops, to things that I consider integrated rich clients, which is slightly different. Integrated rich clients being this multi-screen presence of things on TVs, everything from the in-dash system in your car to desktops to webtops, all leveraging that web as a platform and trying to leverage all of those services and stuff on the Internet. How do we do that? I think that one of the things that we found that I think is actually interesting is the JavaFX family as well as the JavaFX Script allows people to write in a consist and standard declarative scripting language and target those devices in the same way regardless of what platform they're deploying to.

Hal:

So let's pick up on that declarative and easy-to-use angle because as you and I have discussed at various times and in various meetings with Greg Papadopoulos and some of the other technical leaders at Sun, there's this explosion of applications that are content driven. I don't want to say content based, but they really are content driven, whether it's people uploading content or tagging content or sharing it, so everything from social bookmarking and social networking sites, all the way through to user-generated video and user-generated music being shared. Does JavaFX promote that, incent that, enable it? You know, where's the where?

Bob:

Where's the where? Well, you've actually hit on the point, is that we've gone from an environment where the tail is now seriously wagging the dog, and most of that tail is actually about consumers and people who are using the network and all of the data that they not only use, but provide. So what we've gone to is we've gone from a place where largely the corporate entity dictated what the consumer did and we provided sort of static HTML and text to them. And now we're basically looking at a world of, you know, the My Space and You Tube and places where user-generated content is largely driving the applications. We've gone to a model now where by the richness of the client itself is something which is beginning to lag the desires of the consumers to actually present all that data. And so, yes, we've seen things like the rise of the rich Internet application today in the form of things like Ajax, but Ajax itself is still largely limited, right? I mean, there's still a lot of sort of functionality and behavior you expect in your client which just isn't there. For instance, obviously people have seen in the last week or so, Google Gears about this notion of persistence and adding persistence into web applications. But, of course, that's only the beginning. One of the things that, you know, we demonstrated at the keynote at Java I was some work that Jerome Dochez is working on which is GlassFish V3, where not only are you talking about persistence; we're talking about processing in the form of a 100K application server kernel that could easily be bound into a lightweight yet rich client even on a mobile phone. And so things like that where I think that we're going to start seeing some rather interesting leverage of web assets, not only services, but content as well.

<http://blogs.sun.com/innovation>

Hal:

And I think the industry has kind of been stuck with this notion that something's always a peripheral. At the Morgan Stanley Data Center Symposium earlier this week, the networking companies see servers and storage as peripherals of the network. The storage companies see servers as peripheral of the disk, and all of this implies some sort of ordering of one thing being the center of the universe and something else being the consumer, and what you just described is really a leveling of the playing field. It really is – everything is a pier on the network. And as Greg likes to say, producers and consumers and you can't look to some specific ordering here. So the more capability we pack into the client, the more interesting and possibly the more useful applications become.

Bob:

It certainly is a much more holistic view. I mean, I think that if we take a look at an old catch phrase: If network is the computer, well the computer's the network too, right? I mean, it's everything. It's not only the client, but the database, the server, and all the services that are packaged in there. That's what's interesting. And, you know, this notion that from a locality standpoint, they have to be at endpoints of the network is kind of silly if you take a look at sort of modern environments.

Hal:

Well, the endpoint assumes that you're at the one point where you're only consuming. You're at a leaf node and, again, whether you look at your favorite social software site or anything else that involves linking multiple applications together, it's very hard to figure out what the terminal point is. Most of these things are saddle points between different sets of applications.

Bob:

Exactly, and I think you pretty much hit on it. I mean, this philosophy and the way we look at things has got to change dramatically. So the interesting thing about when you're sort of circling around and taking a look at JavaFX and the JavaFX Script is one of the interesting parts of that is how do you actually create those rich and compelling clients? Today, if you're trying to do this in Java, you know, you're sort of stuck doing swing. And if you've actually got the time, then you can start worrying about things like 2D and 3D. This notion of being able to take a declarative scripting language, which is a domain-specific language, specifically targeted at creating those Java UIs, that gives you an ability to all of a sudden dramatically improve and optimize your development experience such that what used to take maybe months to develop, right, in maybe sort of klutzy-looking sling apps, now you can do in rich 2D and 3D in a matter of weeks. I think that coupled with some interesting design tools that are going to be coming up pretty soon from Sun, I think, are going to make it really, really interesting for not only developers but content authors alike.

Hal:

And this has been the motivator, I would say, for almost every interesting shift in applications, whether it was Tickle TK in system administrators to go create graphical tools, or the rise of interest in things like Ruby and Rails as providing declarative and simple frameworks just to reduce the [inaudible] as doing something like Hello World. And you've certainly hit on here how you make the content-rich applications easier to go build. And the easier they are, the more interesting they'll become, I think.

Bob:

Right. And like you said, this is a cycle that essentially gets replicated over and over and over again in our particular area of expertise and interest, which is whether you look at from a UI perspective, whether you look at it in terms or scripting languages. I mean, heck, even object-oriented programming in general. You know, one of

the ideas of course is to promote reuse and make it easier to rebuild larger and more complex applications leveraging other people's stuff. And I think that we're definitely sort of continuing that trend, but now we're applying it to the UI.

Hal:

And you talked about reuse, and I have this funny view of reuse in that most of the things that people look at reusing, they run a while ago. The look at the code and go, hmmm, do we really want to reuse that? That thing that people are interested in using is applications on different delivery points. Now, they want things to work at least somewhat consistently or a somewhat consistent feel to them as they go from device to device. So we talked about "write once, run anywhere." What does JavaFX as an environment do for the anywhere part?

Bob:

Well, let me point at one of your earlier statements first before I answer that, and one of them is that, yeah, you're right. So if I'm going to reuse my own stuff, I typically don't. Maybe that's just sort of an engineer's mentality. But if I have to reuse somebody else's stuff, I can't really demand for them to change it. And so this is sort of the whole basis for a lot of the service-oriented architecture stuff out there today, which is if I don't like the way the Google Map API works, then I can't really affect too much change on Google. I'm just going to end up using it. But the fact that they're using some sort of reusable API that I can leverage, I can build a better app on my side. That make sense?

Hal:

Absolutely. And I think that what you've done is strike a balance between some of the major transversing which is open source where everybody feels they have, and certainly do have, the right and the, uh, requirement to go and change and modify as needed. And those services which are deployed by service providers where you get an API and there's no ability to change them. And what you've done is strike a balance there of, okay, what do you do in that regard? How do you combine those things together?

Bob:

Exactly. Now the answer to the other question, you're talking about how does JavaFX Script move us further along the path of "write once, run anywhere." I think one of the interesting points that you pointed out earlier was this notion that if you're going to build an app on the desktop, it's a different thing that you're building on a phone, on a TV, on a server, and there's a number of different reasons for that. One of them is obviously that the platform is different and the APIs you expose are different; therefore, you have to essentially rewrite portions of your APIs to do so. The web, in the form of HTML and so, was a large leveler in this notion that I could write an HTML client and, at least in theory, it would run on all these desktops, disregarding, let's say, differences between IE and Firefox. But in general, you write something in HTML, you largely know that it's going to run and be able to be usable in all these different browsers. The issue there is that, of course, a browser is a lousy interface on a phone, and anybody who's really tried to use it on a phone has most of the time been fairly frustrated because you don't have the sensor to screen real estate or control over that interface as possible. However, rich applications are actually eminently scalable to small things. And so if you, as the developer, can write to the same set of APIs, at least at the UI level, between whether it's on a desktop, a TV, phone, even an in-dash navigation system, I think, there you're starting to achieve some of that "write once, run anywhere" promise. If, over time, we start evolving the Java platform to follow the same sort of model about creating rich and somewhat standardized interfaces and APIs, I think we can actually extend that even further to, let's say, not only from the UI perspective, but everything from accessing data, accessing services and integrating those things together into a compelling user interface.

Hal:

So what do you want developers to think about? What's your call to action for them?

Bob:

Well, my call to action right now is fairly simple. One of the things that we really need to do is we need to get the company in general to start thinking about JavaFX and what its capabilities and what its promise is. And so I think that there's sort of two sides to that. One of them obviously is become familiar with it. So we have a Java.net project out there, openJFX.net, which is also on Java.net. Go there. Take a look. Read the docs. Try out JavaFX. And even more importantly, start populating the forums and see that we at Sun are all sort of living the JavaFX lifestyle, if you will. From the other side of it, take a look and start thinking about how best are we going to be able to leverage JavaFX in a wide variety of areas, whether it's this move to creating richer and more compelling user interfaces on desktops to phones. Especially if you're out there talking to customers, I think one of the most important things is as a result of knowing what JavaFX is and as a result of participating in the openJFX community, bring that customer feedback back into Sun so we can evolve this and make it something that's truly useful for customers.

Hal:

Great. Well, Bob, as always, a lot of fun talking with you. And fortunately, we get to do this through the wonders of audio recording technology so I don't have to feel short sitting next to you for a change.

Bob:

[laughter]

Hal:

Thanks for being our guest here. And you've been listening to another episode of Innovating@Sun, and I'm your host Hal Stern.

Announcer You've been listening to Innovating@Sun. Join us next time for the latest in innovation from Sun Microsystems only on the Sun Microsystems Podcast Network.