Lotus. Developer Domain
The Technical Resource for Lotus software
LDD Today

Level: Beginner
Works with: Notes/Domino
Updated: 02-Jan-2003

Getting platform specific with Eddy Bell

Interview by
Tara Hall

UNIX, Windows, LINUX—Eddy Bell knows them all. This month we talk with the lead architect for the Domino platforms about which platform he prefers, what to expect in Domino 6.0.1, and whether or not there will ever be another Notes client for UNIX.

Tell me about your position and what your responsibilities are as the lead architect.
As the lead architect, I have to be an advocate for all of the different platforms we run on, including new platforms, to make sure that none of the platforms is ignored. I also am responsible for all of the interfaces between the Domino code and the platform libraries—the operating systems.

As an advocate, does that mean you shouldn't be biased about one platform or another?
Exactly.

So you don't have any platform preferences?
I have my own preferences. I come from a UNIX background, and I really like the UNIXes. But I also strongly believe I can't yell at the NT-centric people and tell them that you have to take care of UNIX unless I do a good job of taking care of NT. So I want to get as much out of every platform as I can and strive very hard for that. I make sure that our whole team takes that attitude.

What is it about UNIX that you like so much?
The richness of functionality that's available, the stability in UNIX. A lot of the things that happen in the NT world are unacceptable in a UNIX world like the blue-screen of death, for instance. Or having to reboot, that's a problem. All of those things are unacceptable, and for the most part, they're also unacceptable for enterprise customers now running Domino. So we have to make sure that we don't inadvertently cause any of these blue-screens of death on Domino. We make sure to keep the machine running and to give it the tools to monitor the system, so you can avoid that issue.

Do you have a certain flavor of UNIX that you prefer, Solaris, AIX, or HP-UX?
Each one has its strengths. For instance, we were recently doing some work on HP-UX, and HP traditionally had a compiler that was so difficult to use that developers avoided working on the HP platform. HP has since evolved their compiler from one that people avoided to one that is not only pleasant but has features that none of the other compilers have. For instance, when we didn't know how to solve the problems that we had with a build across all platforms, the HP-UX compiler not only told us we had a problem, but suggested ways to fix it. And someone fixed it. So for that reason, I really like HP-UX.

For debugging certain types of problems, I want Solaris. For doing real world performance measurements, I like AIX. For other things, I like Linux. It just varies, and in fact, what we are be doing is looking to the strengths of each platform. A lot of the problems that we encounter exist across all platforms. We, therefore, have a choice of which is the easiest platform or platforms on which to isolate the issue and to come up with a resolution. We often use several different platforms to gain a better understanding of a problem. In the end though, we validate the solution on the platform that the problem was reported on.

We get a lot of feedback from the LDD audience asking for a UNIX client. Are there any plans to bring back a Notes client for UNIX at this point?
This has been a long-standing problem. I've been with the product from way back when there was a UNIX client, and even though I was unhappy with the UNIX client going away, I understood the reasons why it had to go away. And I reluctantly agreed with the reasons. Those reasons still exist. The biggest issue is that the customer wants the features and functionality that are in the current Notes client, and they equate that with needing the Notes client itself.

There are several projects currently going on that will give us a much better Web-based client that will give you, to a large part, the features and functionality that you get in the regular Notes client. Using a Web-based client actually has a lot of advantages for the customer. It's much more cost-effective. Administratively, it's much easier to maintain because you don't have to do an installation and maintenance of every individual system you're running that client on. It can be a centralized and centrally administered product, which for large enterprises is much more what they want. It lowers the total cost of ownership significantly.

How about a Linux client? Any plans for that?
There are a lot of people who would love to see a Linux client. There is a lot of momentum, but it all comes down to dollars and cents. If there's enough demand for one that can't be met by a Web-based client, then we'll do it. It's largely that we want to make money, and we have to look at the balance. It would be a monumental effort in the current code base to support a Linux client because all the GUI portions are basically Win32 GUI, OLE, ActiveX. It would be very hard now to go back to a UNIX-based client.

Before we had an abstraction layer that made it a little bit easier, but that abstraction layer was preventing us from responding to the Exchange client and adding new functionality on a timely basis. Some people had quite a bit of success on Linux running the standard Domino client under an emulator like WINE. Maybe the easiest way to address their issues is to come up with some method of certifying it by starting off with the most popular emulators, and then saying, this is a supported configuration. Then from our perspective, it's a testing effort rather than a development coding effort. It's up to the people who are using it to raise the demand to a point where I'd be more apt to respond because the benefits will outweigh the costs.

What are the development issues for developing clients that are UNIX-based or Linux-based?
The issue is that in the current development tree there's only one tree across all platforms. So for any instance of the file or area of functionality, there's only one copy. The rest of Domino works very well, but because we're going to platform-specific interfaces for the display-based client, it would be a huge development effort to change those. If we did that, then you'd have to look at how much that would destabilize the existing products. And the testing efforts are very large. If the demand gets high enough, we'll do it. But until then, we can look at lower-cost alternatives and at the root needs of the end user to see if we can meet those needs with other types of applications, preferably Web-based applications, at a much lower cost.

If you had to coach a customer in how to choose a platform, what would you recommend that they look for? What factors should customers consider when choosing a platform for Domino?
What is their in-house expertise? That's the biggest factor. The biggest cost for a Domino server is the administrative cost of it. If a company has, for instance, a strong Solaris background, and they have resources there that really understand Solaris, go Solaris. If NT, go NT. There is no one solution.

It also depends on how you want to operate. Maybe you want to consolidate down to a few really large servers, or you want a distributed environment with lots of smaller servers that can be geographically distributed. Or it may be that you want a mix. Take a company like DaimlerChrysler. In the United States, they're a Solaris house, but in Europe, they're an AIX house. One of the nice things about Domino is that we don't make the platform decisions for you. We let you decide what you're most comfortable with. A lot of the big corporations have invested a lot in their relationships with vendors. They're comfortable with that platform, and they know how to deal with it. It's important to keep that going.

Do you find that larger customers tend towards a certain platform, while smaller customers tend towards a different platform?
Yes. Larger customers tend to look at dollars—what it's going to cost. They're very concerned, for instance, if you get a new Notes client, what the training cost are going to be for training for all their end users. How hard is it going to be to migrate? They're very concerned about that. They also look at hardware costs and reliability. They ask, can I guarantee that if I have a failure of my hardware that I have new hardware there in six hours? For some platforms, you can get that; for some, you can't.

If you look at some of the large installations, they have thousands of users who go down whenever their hardware goes down. So a failure is very costly. If they lose an hour because the server went down across several thousand people, that's a lot of money. They put more emphasis on reliability. They also do more with redundancy. They tend to use clustering, both Domino clustering and hardware clustering, so that they have higher availability. The back-up machines are sitting ready to kick in at a moment's notice.

There are some interesting things that customers have done in that area. I believe the California Department of Transportation designed their environment around earthquake disaster tolerance. What happens if this center goes down because of earthquakes? How can I off-load to other areas?

The Department of Education in Hawaii has all these islands with smaller requirements for the Department of Education, and they want centrally administered servers. So that's dictated the platform for them. Up until recently, it was very, very difficult on a Windows platform to do any type of remote administration. The Department of Education, for instance, use Solaris.

People choose platforms specific to their needs. The more costly downtime is to them, the more they gravitate towards the more reliable hardware, more reliable OSes.

Small businesses are more flexible. The downside of a crash is not as costly in a single hit. They balance the cost of a server crash versus lack of performance or other issues. Small companies sometimes hedge on how they do back-ups, for instance. They aren't as adamant in making sure that they have a very fine granularity back-up schedule that's very reliable. Big companies don't have a choice. So each company should be able to choose for its own needs. One of the nice features of Domino is that it lets you do a lot of this, so Domino isn't the major factor when you choose one platform or another.

Do Domino features like scalability and capacity vary on different platforms, or is Domino pretty much the same across all platforms?
It depends on what you mean by scalability. In a single instance of Domino, or a single partition of the Domino server, they're about equal. But, for instance, if I get a really large UNIX box, it has more power than the biggest of the NT boxes, so if I put a lot of partitions on there—on a single box—I can run a much heavier load. The major cost, though, is the administration. The administrative needs are pretty much based on a core partition environment. You still have about the same cost. You do have reduced hardware cost if you can put more on fewer machines. We're looking into the future where the machines are actually laid out differently. We're looking at, for instance, mass storage. We're actively looking into what we have to do to play in a NUMA (non-uniform memory access) environment, where you have many computers functioning as a single computer. If you look at some of the big NUMA environments, it's a little bit mind-boggling because you can have 128,000 processors and thousands of terabytes worth of RAM and who knows how much disk space. It's a real challenge to figure out how you could run in that environment and do a really good job.

In the upcoming point releases of Domino, what should customers expect for platform support?

I believe Domino 6.0.1 is on track for Solaris 9. We have third-party components that we have to make sure support platforms we're going to. We try as a rule never to "desupport" in maintenance releases, so that, for instance, if we started shipping Domino 6 with Solaris 8, at the end of the life cycle, Domino 6 will still support Solaris 8. The idea is that it's a really bad thing to force a customer to upgrade their OS to take a maintenance release from us. So far, we've done a very good job of that. There's only one instance when we dropped support in a maintenance cycle, and that was for the Intel Solaris. We worked very carefully with the customers and with Sun to manage that. It was not a decision that was lightly made. A lot of work went into verifying that we were not going to strand customers.

In what capacity do you work with the Domino for iSeries and zSeries teams? Are you involved at all in their planning?
Yes, we work rather closely with them. We keep each other informed about what we're doing. For instance, we were playing around with some changes in the R5 code stream. That code stream is yielding very significant performance improvements on AIX and Solaris, which should significantly improve the capacity of those systems. We let the zSeries and iSeries teams know right away that we're doing this, what the changes are, so that they can deploy the same changes and start measuring what the performance gains are. Likewise, when they add stuff in, they keep us informed of the changes, and we try to utilize it as much as possible. We really like to leverage each other's work.

In the upcoming point releases, are we going see any improvements on certain platforms?
You will definitely see improvements on certain platforms. Most of the performance improvements that we see will be cross-platform. So you'll basically see it across all platforms. And like I said, we're working on some stuff now that includes relatively small changes that make a significant boost in performance. We're evaluating new hardware that's coming out and what we can do to better utilize that. That could yield performance improvements. We have to balance what we're doing against the testing resources that we have. We can't just go in and make major modifications. We have to be very selective in what we're doing. But I would say that we're going to see an ongoing performance improvement effort.

How do you determine when you're going to introduce a new platform?
The new platforms, including new releases of OSes, are handled by the product managers. They come to us to find out what the issues would be if we introduced a new platform—get an idea of what the cost would be—and then they make the business case that it makes sense to do this. So it's a little bit different than it was previously. This is actually probably a better way to do it to prevent us from making mistakes wasting resources where we could have saved them and applied them in a more cost-effective manner.

Previously, were you doing it without product management involvement?
We had some product management involvement, but basically we would have customers calling in and saying they really want a platform supported. Lotus Support would talk to them, then we'd do an evaluation in development as to what we thought the risk was. That has a lot to do with what the history has been for the platform. Then we would make the decision and apply the resources to do it. It was the squeaky wheel that got what it wanted, but now it's really down to dollars and cents.

New platforms involve a very significant testing effort for us. And that's why decisions to introduce a new platform are not lightly made. We have new platforms that we're considering for future releases that most people would think are existing platforms, but it would be doing 64-bit versions of Domino. If we did one for AIX or Windows or Linux or Solaris, each of those would have to be treated as a brand-new platform. So we'd have to have the justification for the resources, both testing and hardware resources, that we need.

When a new platform is released, how long do you wait until the platform has proven itself before you make the decision that you'll support it?
We wait until the customer says, I'm convinced that this platform is where I want to go, and I really want your product on this platform. It builds the case there. Most of our customers are enterprise customers, and they don't usually make those choices at a whim. Linux was our most recent platform, and we saw a lot of the momentum for it, but we decided to just kind of do a skunk works project with it. Then when marketing said that they'd really like to get on the band wagon, we had a running version. We still had to make it a product, which was a significant effort because we didn't have the testing resources to bring a server up. We could cover basic functionality. The testing uncovered a lot of issues, but it was well worth it. It was very rewarding for a lot of people. A lot of people were excited that we were jumping on the Linux band wagon.

How difficult is it to support so many platforms in one release?
The biggest issue is testing. We have a pretty good handle on how to do the development, where developers are responsible for areas of functionality in the code. They're responsible for that across all platforms. We've extracted the majority of the platform-centric areas of code, and those are areas that my group owns, so that they write to a

standardized interface. If they're writing on Windows or on AIX or Solaris, it doesn't matter. They don't really have to worry. We've set up environments that make it very easy for people that are not UNIX literate, for instance, to build the product on UNIX. In fact, for somebody who's never been on UNIX, he can build the whole product with less than a dozen commands from his PC workstation and build thirty thousand files to something that he can run. We even have in the works a next generation of these tools that brings it down to two commands.

The idea is that we make it easy and make it extensible for people. For instance, the next generation tools also include the iSeries and zSeries. The idea is that we want the developers here at this location to be able to actually verify their changes in their code on the iSeries and zSeries without actually having to know those platforms.

Are you involved with helping to choose platforms for other products?
I consult with different groups as needed about the issues that they will face and try to get involvement from the different vendors to help influence the decisions. For instance, one of the things that we've been very successful with is setting up some relationships with some of the vendors. With Sun, for instance, we have an excellent relationship. We do a lot of projects together and really look at issues together. That's been ongoing for years. We have an excellent relationship with the AIX development team too. We're doing quite a bit of work with Microsoft working on common issues for common goals. By working together with all of the platform vendors, we offer the customer solutions to issues, not finger-pointing.

Linux is a little bit harder because there isn't one place to go. Do we support SuSE? Do we support Red Hat? We're constantly trying to find a better way to do that. So we work within the IBM Linux community and participate there. In fact, the Linux community is developing new functionality that we plan to use in Domino. In 6.0 and 5.0, Linux was the only server platform that did not use thread pools or threads for connection. That was due to limitations in the OS. The new interfaces that we're getting into Linux will allow us to have that functionality and on the high end, will definitely benefit the Linux community on Intel. If we do a zSeries Linux, they'll have a big benefit from it as well.

Are you involving the customer early to help determine some of those platforms for these other products?
Yes. The customers are involved in defining what the product is. I hope that we'll see more customer exposure, for instance, at Lotusphere, and a lot more feedback from customers. Right now, just the logistics are such that we're looking at really large customers for input because it covers more of our install phase. We also want to listen to the smaller customers, and Lotusphere gives us a good avenue for customers first, to be exposed to what we're trying to do, and second, to give their feedback. The forums are very useful for feedback too. A lot of people inside Lotus read them and listen to what customers want. We take that into consideration when we make our decisions.

What criteria do you use when you decide that you're no longer going to support a platform, such as OS/2?
Part of it is when the platform has stagnated, like OS/2 did. It wasn't developing any new code, and we're an evolving product, taking advantage of new technologies, like Java and new changes in standards for C++. When platforms stagnate, it becomes more and more difficult for development to encompass the lack of functionality and features that the platform offers. Usually when you have an OS that's stagnating like that, you have people migrating away from it.

In Europe, we had a lot of people who were using OS/2 because they are, in general, very anti-Microsoft. Their choice for migration was Linux. So you have a very large migration there. We try to reflect that in our products. Then we try to give a lot of advance notice that we intend to do that, to give customers a chance to raise objections. With OS/2, we got a lot of objections from Europe, and in fact, we kept OS/2 alive a lot longer because of those objection. But we constantly have to remember that the customers who wanted it make our existence possible, and we really need to deliver what they want, not necessarily what we'd like to do.

If what the customer wanted weren't an issue for you, would your platform choices differ? Are there platforms you wouldn't support, or are there platforms you would choose to support?
I think we have a very good mix now. I would personally like to see HP back in the mix. I think there's a very large market that is mostly untapped from lack of the sales teams coming up with a method for penetrating those markets. I think other than that we're very well along with what's current. We are looking very hard at the 64-bit platforms because we consider those new platforms. That opens up new worlds to us.

Right now, in the 32-bit world, we're basically up against the four gigabyte address space limit that we have on all platforms. We'd really like to go beyond that, and we're in the process. We will be doing it—it's just a question of when and with what code stream. We may actually start a 64-bit board for an Office 6.0 code stream, and then decide that it makes sense to have this come in the 7.0 stream. But it's a balance between the market needs and our resource availability.

How much of Domino's selling on a certain platform helps to advance that platform?
Significantly. All of the vendors have turned out to be quite responsive to our needs. We even go so far as to areas of our code we actually share with them, of the platform-specific code, saying this is what we're doing. Tell us how to make it better on your platform. And we'll do it. This is the platform-agnostic view: We want to do the best that we can on every platform, and let the nature of the platform, and the desire of the customers dictate which platforms they end up with.

Is that also true for our relationship with Microsoft and their platform?
The Microsoft platform is a huge market, and we want to play in it. And as I see it, as time goes on, which hardware platform you're on will make less and less of a difference. It will be a question of WebSphere, .Net, or J2EE. It's the actual OS vendors who are going to be less important to the customer. Customers will start choosing vendors for whatever benefits the OS platform can offer them as far as system costs go.

Let's talk about your customer experience.
One of the things that I like very much is listening to the customers and talking to customers. Unfortunately, often by the time I'm involved, they're not very happy. Maybe it's been a difficult problem that hasn't been resolved as quickly as they wanted. But I still really like being involved with the customer and going to Lotusphere to talk to the customers and to hear firsthand what their problems are. The rest of the year I listen to product management, who's gone out into the field. There are several layers in between, and you lose a lot in the translation. I like to really hear what the customer has to say, and it also gives me an idea to play across to customers. What if you could do this? And I feel that that's a very important part of the product determination. You can't just ask the customer what they want. You have to tell them what they could potentially have, and then ask them what they want.

We came up with a Domino 6 feature that's a Java-based server console. When we first wanted to get input, we talked to a bunch of our customers and said we're redoing this from scratch, so if you could have anything you wanted, what would you want? They kept asking for small, incremental improvements. We said, no, no, no. Think big. Think big. We went on and on with this. And they just weren't doing it. Then we said, OK, this isn't working. Let us tell you what we're thinking about doing. As we drew it up on the board, their eyes got big and they said, Wow, give us a few days and we'll tell you what we want. Then they came back with a whole list of features that they wanted. That's a much better way to define the product. So it means that development has to really do a good job of educating product management and the field as to what the potentials are for things that can be done. I think that's where we get the exciting new innovations in the product.

ABOUT EDDY BELL
In 1994, Eddy Bell left Sun Microsystems, where he was working on PC emulation products 386i (roadrunner), SunPC, and WABI, and moved to Lotus. At Lotus, Eddy was the Project Architect/Leader for porting Notes/Domino from Windows to UNIX. He is currently the Platform/OS Services Architect for Domino.