Interview: Domino Web server team

Interview by Betsy Kosheff

[Editor's note: This interview resides in "Notes Today", the technical Webzine located on the http://www.notes.net Web site produced by Iris Associates, the developers of Domino/Notes.]

The Lotus Development campus in Cambridge, Massachusetts, is dominated by the towering Lotus Development Building overlooking the Charles River. Across the street in two less imposing structures, the engineers live. The older of the two facilities, known as Rogers Street, houses some 700 developers, as well as parking for most of the 2,000 employees in Cambridge. The building itself looks like a garage. Inside, people write code pretty much the way everybody else does. But what you might not expect, is that even in a company the size of Lotus, now an IBM subsidiary, software like Lotus Domino can still be developed -- from concept, to design to architecture and programming -- in much the same way that the earliest PC products were developed, by a small group of people banded together in a garage with a common idea. The core Domino team is less than a dozen people, but the initial concept began with a couple of people batting around the idea that Notes was the ideal server for the Web. We caught up with several members of the team to talk about the technical details of Domino, and to find out their plans for future enhancements.

Who came up with the Domino name? What does it mean?

Dennis Cunningham
I proposed it as a code name, but it was Julio and Miguel Estrada who provided the impetus. They were engaged in a weekend long match of Dominoes with their father and uncle, and they were describing what it meant to be involved in this game, how intense it was, and how seriously everyone was taking it.

Julio Estrada
In Spanish, Domino is pronounced much more emphatically "DomiNO," and in the game, the person who places the last stone on the table wins, or dominates. It's typically done with a very violent shot on the table. It makes a big noise and all the pieces kind of jump up and down. I guess in a way we were hoping Domino would do that in the Web server market.

What were your initial design goals for the product?

Dennis Cunningham
We started off thinking about rewriting Web Publisher with a particular focus on making it more interactive. We had some specific things we wanted to do: We didn't want to use CGI; we wanted threads to go

straight through from the base Web server to Notes. Our overall goal started with a sense that, let's just build a product that works completely interactively, instead of on a publishing basis. The big question at the beginning was whether to do this from scratch, or use the Web Publisher as a base, and Miguel and Julio wanted to write it from scratch.

Miguel Estrada
Even when we were building Web Publisher, the idea was always to do things dynamically. It turned out that Web Publisher ended up being a publishing type of product, but the idea was based on research that Paul Haverstock had done with Irene Greif (Lotus Director of Workgroup Technologies) to build a server of Notes information in realtime. This was over two years ago. We pretty quickly cobbled up a prototype using the Web Publisher code base so that you could get realtime pages from Notes documents, and at that moment we had something that was really what we wanted to have -- a dynamic server. That was February 1995, but it was put on a shelf until we finished the Web Publisher product. This was done because it would take a lot more time to finish the real-time product than to finish the publishing product. Looking at it now, I wish we would have chosen to do the real-time product. So, it was way back then that we started to have some of the ideas that would later become a dynamic server called Domino.

What happens to Web Publisher now?

Colleen Griffiths
All of the functionality (and more of course) that was available via the publisher is available in the Domino server now. The Web Publisher will not be enhanced further.

Were you thinking of building from Notes, or starting from scratch?

Julio Estrada
We were building from Notes. In general, what has guided the design goals was the thinking that: "Here is a Notes server, and there's a lot of functionality already in here. Mosaic is very popular, so, how can we expose the functionality of Notes -- at least the basics or as much as we can in a short time -- to Web browsers like Mosaic?" So we started by identifying which objects within Notes could be exposed to the client -- views, forms, databases, the server, those were obvious -- and what actions or operations in those objects could be exposed to the Web browser to bring it more on par with the Notes client. Of course, this was both easier and harder to do than starting from a functional spec, because Notes is a living spec. So we looked at Notes and said, okay it can do all these things, but which subset can we expose in the time we have -- what is most important to the market? The Notes server, databases, views, documents, forms, data navigators and agents, and of course, the security. That was a big challenge, trying to marry the Notes security to the Web security model because they are so different.

Was there a lot of discussion around which pieces of the server you would expose to browsers?

Colleen Griffiths
Yes, because people were concerned that such a strategy would undermine the Notes client business, so it wasn't a popular thing to be doing, even as late as last summer of 1996. We felt the most important thing Lotus could bring to the Internet space was the power of Notes applications, and to bring that to everyone -- whether you wanted to interact with the applications from a Notes client or a browser. It was our teams' job to get across the view that there would be a change in the browser business and that thin clients would get bigger and richer, so if there was any "negative impact" on the Notes client it would be a temporary condition -- if you look at where Netscape is today, that statement seems obvious, but in early to mid '"96, this was a risky point of view.

Paul Haverstock
One of our big supporters was Ray Ozzie -- he was very supportive of what we were doing. People knew that the business needed to go to a server model, but at the time, our pricing was not matched up to that.

We were operating on a client-based model, and nobody knew quite how to transition the business accordingly.

Dennis Cunningham

The development people at Lotus were using the Web long before the general population of Lotus, and certainly the business world in general, so we had a good view of how it could change our business. It took people internally different amounts of time to understand how this was going to affect Notes. Before we first showed Domino externally, we had been living for a long time with the notion of Notes databases and applications being served up to the Web, so that seemed very basic to us. What we wanted to do was deliver some really cool stuff, not just serve documents.

It sounds like the principle challenge for the team was more a psychological battle to get the company in agreement, rather than a technical one?

Colleen Griffiths

Absolutely. To give you an indication of the initial mindset, at one point, we didn't have an external handle for the product, so people were calling it Notes Release 4.11. That seemed like a blip on the radar screen -- a mere point release, yet it was so much bigger than that. So, when we went to our first open beta, we created a site with a different personality and a new name called Domino. At that time, there was a stigma in the market -- Notes was just uncool to the Web audience, but we knew they'd think easy to build dynamic apps were cool so we just went for the different name to help us make the point.

Dennis Cunningham

At one point, we thought the name was going to be HTTP Services for Notes, and Domino was so much more than that, but then John Landry came to the rescue and said "Hey, this isn't some driver." He was our beta site, our benefactor, our harshest critic, and he was the one who started telling people it's an application development platform for the Web.

What was the competitive landscape?

Miguel Estrada

A year ago, we were getting this criticism that Notes is too big and too proprietary, which I believe was fueled by Netscape positioning itself as the best product for intranets. It was at this point that we said, "Hey, they are really going after our food!" They're not just trying to be an Internet product for consumers, they're going for the heart of our IT business, the intranet. Lotus' key customers were seeing the alpha release of Domino and they wanted us to use the Domino brand to change people's perceptions. That's when marketing kicked in and it went from, let's make Notes a Web server, to let's create a whole new product identity.

What was the most difficult technical challenge you faced and how did you solve it?

Ned Batchelder

We were concerned about quality control and time to market, so we could get the product out in June. And performance was the biggest thing, making sure that we would be able to serve pages fast enough.

It's a server product, so we kept running across the issue that we had no control over the client UI, or our client design environment. For example, we could not add a check box to the infobox, or if we had a cool feature we wanted to implement, such as a view with no Domino formatting at all, just raw HTML concentrated together, we couldn't control the user interface. This led to some design constraints.

The other major issue involved the limitations of Web browsers. Look at the richness of the gestures you can make in the Notes user interface -- you can click, you can double click, you can shift click, you can use keyboards and menus -- and then you see that in a Web browser, all you have is a click. There were a lot of things we could have done, but didn't because of these limitations. For example, to view navigation

in Notes, you can expand a row once, or expand a row completely, or expand the whole view completely, because there are different key strokes for those operations. On the Web, you can't expand a row completely, because to do that, Domino would have to display a clickable thing on each row to do it, and that would be too much visual noise.

Dennis Cunningham
I was amazed that our architecture lasted for the length of the product development cycle. It's not very often that you start with an architecture and it sees you through even the first release. You can't conceptualize everything you're going to need to do, and all the feedback you're going to get that's going to make you change things, and still have that architecture hold up. Partly, it was that the problems stayed relatively simple and didn't get out of hand. But I think the architecture was good -- it's an object-oriented architecture which really held up well and will hold up well for another year or so.

How easy would it be for someone else to create Domino?

Paul Haverstock
There were other companies who were building different sorts of gateways between Notes and the Web, and at one point during the initial development of the Web publisher, there was one company that was building more dynamic translation than we were. But they didn't survive, I think because this group was able to execute better and faster, more consistently over a period of two years.

Julio Estrada
There is no question that our competition could do what we have done and better. The worst thing we could do is rest on our laurels. They could have done what we did, but it was so much easier for us having had this system in Notes which already addresses the principle client-server and workgroup problems, and then add the new standards. So we didn't have to build the entire infrastructure, the object store, replication, security, messaging, etc. And although ActiveX, for example, is object technology it is still very difficult to use, requiring highly skilled developers to put it all together for the customer. Of course, this spells opportunity for developers, but it's a lot of work. Domino benefits so much from having all the objects, the forms, everything all built in. The bottom line is, what the customer really wants is to create an application that solves a problem, and Domino is the quickest way today for them to build an application. If you put one team of developers on one side of the table using Microsoft technology, and have another team using Domino, the Domino team will beat them by days, weeks, even months.

Ned Batchelder
Part of our success has been that the team started small and stayed that way. It might surprise people who think we are some big unwieldy corporation to know that we started Domino in December and shipped in July, and there were only three engineers working on it for the first three to four months, and only three more for the next three.

I think the reason that Domino is so good at solving application problems is because it's designed to follow the feature set of a product that's had four and a half releases over 8+ years, and all of those releases have been tuned based on customer feedback, on what they need on their intranets. So it makes sense that we have the best intranet Web server.

How do you answer criticisms that Domino is slow compared to other HTTP servers?

Paul Haverstock
We answer that we're doing something different. We're dealing with live objects, not dead files.

Harry Peebles
We're dynamic. When we're serving up Notes pages, so a lot of things are being figured out and laid out at that moment. Formulas are being evaluated, workflow is being determined, all kinds of things are happening, which is very different than just sending a file over, as HTTP servers do. That makes what

we're doing very different, and therefore, harder to compare with traditional Web servers. However, the server that we're employing can do all of the things that HTTP servers do, and you can do it right now, from Domino 1.0. So, if, for example, you have a whole series of HTML applications laid out on your Web site, you can serve them up with Domino.

Miguel Estrada
Harry touched on an important point which is that everything is presented dynamically and is not flat static information. You can take that and evolve it to say that what we're really doing is providing the mechanism for creating UIs for applications. So, just like a typical Windows application, an application comes up and it has Dialogues and it has Windows, and everything is generated dynamically in the application, based on how you interact with it. It's the same way that we're approaching this new generation of HTML applications. What's important is the UI for applications that you generate for your intranet, not just saying, here's some information and you can look at it, and that's it. There's a reason why we're not just a file server. You need to generate the user interface on the fly, based on the dynamics of whoever is running that application, and other factors such as time. Domino is an application server and application development environment, versus a database of information that you can only look at.

What technical challenges did storing HTML in Notes present, and how are you going to address this issue going forward?

Paul Haverstock
There is a field type in the .nsf subsystem that is type HTML, and we can store HTML data bits in Domino. The problem is, there is no way for the Notes design client to call it with HTML data; you can only do it with a program through the API. So, we have had to live within the world of what you can do with the Notes design client. But that will change with the next round of clients.

Dennis Cunningham
Functionally, the Notes storage system is dynamic, and basic HTML is not that dynamic. So, for instance, we had a lot of problems with posting modification dates of documents to the Web, since Notes documents are often dynamic, reflecting changes based on time or a given user. This is a somewhat more powerful feature that the Web architecture did not completely handle. If you look at any one of our databases here at Lotus, there's a lot of dynamic information that's up-to-date the minute you open a document. Static HTML files aren't really setup to do that. Notes documents really are dynamic in a real way, not made that way through some sort of server-side kludge.

So dynamic, is the name of the game?

Dennis Cunningham
We absolutely think so. We're just in the infancy of the web, and HTML is great if you're just going to publish your doctoral thesis and that's it. But no one will look at your site twice if they know it doesn't change every day. If it's not up to date, why will I revisit? In terms of building things that are dynamic, that is where Domino shines, because it is by nature, dynamic. You can set up the application quickly including formulas or agents to make it dynamic formulas in etc. and you don't have run across the grain with a CGI program the way you do with other sites. That's because HTML hasn't gotten to the point where it's dynamic, so you often have to use things like Java applets for things that change on a regular basis.

How do you see HTML evolving?

Julio Estrada
This is an important question. As HTML continues to evolve as the open standard for defining how a page looks and its components, Lotus is going to greatly benefit by being at the center of this. For one, we won't need to translate, and two, we believe we're in a position to help better the specification because we have

already addressed a lot of the major problem areas. HTML is evolving to accomplish many of the things that Notes does, so we must help evolve it. This is why we are actively pursuing this.

Dennis Cunningham
It's not just the issue of the HTML format, but the object store going forward. We very much want to be involved in evolving the object store, and we'll continue to argue that Notes is a better model than a flat file system.

There is a big marketing push for "interactive applications" using the Domino server.

What are the plans for exposing browser elements like cookies, user agent strings, etc., so that developers can make their sites customized for a particular browser or user?

Ned Batchelder
You do have access to all of those things through Notes formulas, but we're making it easier by making the functionality more built in.

Miguel Estrada
Right, to give an example, Domino applications can tell which type of browser is in use. The information in cookies is available, so that you can take that information and make some decisions. The challenge is, to give you an easier way to use that information. In that regard, we're in the same boat as everyone else. They get the information and they have to write a Perl script or do some other programming to do anything with it. You can run an agent today in Domino so that, for example, whenever I fill this form and press the submit button, it runs a little program on the server, and it finds out what browser you're using, and what value this cookie has on the browser. And then, it will give you some pages that are customized. You can do that today, it's just not as accessible as it can and will be.

Where is Domino going from here?

Ned Batchelder
One thing we're working on is rendering, so that it will be less of a issue from a fidelity standpoint. We'll give you a design client where you can just deal with the Web in first place, without doing translation. People would rather not convert their Web ideas into Notes, so we'll give them a straighter path where they can just put their Web ideas into the client.

Miguel Estrada
Overall, we want to make Domino easier to setup and administrate. We want to do more Web-centric application development. And more enterprise Integration. One of the big areas we're going to address is Domino integration with relational and other databases and transaction systems. Many of the applications that are needed are those where the data already exists in mainframes or some other database, and certainly Domino feels today as though it requires that you put your data into Notes, but that's going to change with better integration with enterprise systems.

Colleen Griffiths
In general, the next major wave is that the server and client pieces are coming much closer together. The new design client will feel like a really good web product from start to finish, with easier install, a more web-oriented UI, so that anything that feels extraneous is going to go away, for example, things that aren't supported in Web browsers, like inter-line spacing. We also want to more aggressively pursue the ability for people to use other design clients, and we'll support a lot of choices. Plus, lots more on performance, scaleability, and manageability. Particularly on performance, we think there needs to be a new benchmark for a dynamic server so that you can compare apples to apples, and we're working on that. And, our future focus is very much on ease of use. We've got a great application development infrastructure, and now we can deliver even better UI work and experiment with things like stripping out of some functionality in order to focus on the 80 percent rule, making it all more accessible. Obviously, the first vendor that gets to

ubiquitous application development in this space is going to win, and we think we have a really great shot at getting there.

How much of the Notes client functionality can use with Domino and what's the plan to increase that?

Miguel Estrada
In the development of the Web Publisher, this was a question that came up all the time -- people wanted to know: are we developing something to emulate the Notes client on a Web browser or are we developing something to set up a Web site? The answer is that we need to do both. For the installed Notes customer base we need to provide more Notes client functionality, and for the Web application development community, we need to provide more Web-centric mechanisms. We also played with some things like the three-pane UI because there are customers who want a cheaper Notes client, they thought they wanted to replace the Notes client with a Web browser, yet they still wanted to run their Notes apps as Notes applications, with everything being Notes-centric. But the more common model is that people are creating Domino apps that do not necessarily have to have a Notes client on the other end, and therefore, do not have to emulate anything that the Notes client is doing.

Ned Batchelder
Yes, in fact, some people value their applications more if they don't look like Notes at all, and that's fine.

Julio Estrada
I suspect that one day, there will be no difference between the Notes client and the Web client; there will just The Client. There is nothing stopping the Web client from being the universal vehicle for delivering functionality.   The issue of which features of the Notes client go into the Web browser shouldn't exist, because the Web browser will be the client and the vehicle for delivering functionality. So, we will be delivering Java applets to the end user that allow the user to manipulate Domino information, for example.

When are you going to support the ability to create controls and locate them on a 2-d space, similar to what is provided by layout regions?

Miguel Estrada
We do want to provide the ability of doing x/y based element creation, and we'll be saying more about that soon.

Is there one message you think hasn't gotten out about Domino?

Colleen Griffiths
The message about being a dynamic application server is getting out there on the server side, but I don't think people see yet how easily and efficiently you can develop and deploy a range of applications from simple to sophisticated using Domino, versus stringing together a lot of the Web based tools that are out there.

Ned Batchelder
People don't know yet what the full range of things is that they can do with Domino. When I go to the Domino discussion database
(http://domino.lotus.com) I continually find people building things that I didn't know you could do, and making Domino do things that I could never anticipate. This seems to be an excellent indicator of the engineering in the product, whether from the original Notes design or how we're providing it on the web. People have built pages that do things that, if I had been asked, I'd have said you can't do it. Even if we didn't do anything new with the product, I think we'd see a year from now even more amazing apps on the Web, as people understand how far they can take it.

Have you been surprised by the positive reception?

Miguel Estrada
We thought it was cool software and was needed, but we never anticipated the reaction of the market or even the excitement from people inside Lotus. I do remember the team was having dinner one night and we had just come up with a few builds, and we started to think that maybe we have something big here... This was right after we got the work that Dennis did that would let people create dynamic pages based on who they are.

Why do you think people have taken to Domino so readily?

Julio Estrada
One of the reasons for this is that in the past, Notes has been viewed as a package that you get and you mostly use the applications that come with it. Many Notes customers do not create applications and they do not modify what comes with it. Maybe they use mail, discussions, the document library, but most people don't understand all the things you can do with Notes to create applications, because they didn't need to. Now, with the Web, they need to create applications for their intranets and they need to do things nobody else has. It's purely competitive. So they're more obligated to explore all these functions and all that's available to them.

Dennis Cunningham
We just felt that the Notes server is a great server, and if it missed the Web train, people would never find out that it was a great server way ahead of these other servers that wished they could do what Notes can. That would have been a shame.

Julio Estrada
We saw a great apple on a tree, a beautiful apple that we could actually get, and we were very hungry. To get to the apple, we used the shoulders of this great thing called Notes.

Is there an Easter egg in Domino?

Group
We could tell you, but then we've have to kill you.

About the Domino Web server team
Ned Batchelder - Architect
David Boudreau - Quality Engineering
John Chamberlain - Quality Engineering
Robin Chandler - Quality Engineering
Robert Congdon - Architect
Dennis Cunningham - Consulting Engineer
Scott Davidson - Product Designer
Eileen Driscoll - User Assistance
Julio V Estrada - Systems Architect
Miguel Estrada - Systems Architect
Angela Finney - Marketing
Colleen Griffiths - Product Management
Paul Haverstock - Director of Development
Harry Peebles - Principal Engineer
John Walsh - Webmaster
Andrew Wharton - Quality Engineering