



Level: All**Works with:** Lotus Learning Management System**Updated:** 03-Feb-2002

Lessons learned: Fernando Salazar and the LMS architecture

With the launch of the Lotus Learning Management System (LMS) 1.0, we wanted to find out more about this next generation of LearningSpace from the chief architect, Fernando Salazar. We spoke with Fernando about the benefits of the LMS architecture, migrating from LearningSpace to the new Lotus Learning Management System, and his work with customers.

What is the relationship between LearningSpace and the Lotus Learning Management System?

LearningSpace 4 and 5 have been in the marketplace since about 2000. They're Web applications, and they're based on the Microsoft platform, so they use Internet Information Server (IIS), Active Data Objects, COM, and the standard Microsoft suite of technologies. LearningSpace had good success in the marketplace primarily because it was a workgroup product intended for deployments of, say, a few thousand users—maybe ten thousand users. The Windows platform has a lot of administrative and management qualities that are attractive when you're designing a system at that level. So, it's really very easy to set up LearningSpace 4 or 5 and to get going.

Now we come to the LMS 1.0 product. One of the things that we wanted to address there was the capacity limitations of the previous product. Part of that was architecture and part of that was platform, but the end result was that LearningSpace 5 really was just not a good choice if you had, say, one hundred thousand users or more. What many customers want to achieve today is economies of scale. They want to make enterprise-wide systems that are common across their entire organizations and to not allow a proliferation of many different workgroup-type deployments. We took as one of the key goals for LMS 1.0 the ability to support an entire enterprise's learning needs. We defined that as user bases of one hundred thousand or more with at least three to five percent of the user base concurrently active. Those are the performance targets that we wanted to address with LMS 1.0.

I don't really want to get into the technological politics, if you will, between the different platform choices. I think it is possible to create systems that support that load on both Microsoft platforms and non-Microsoft platforms; but for IBM, the Microsoft platform is not the strategic one, rather the J2EE platform is. The WebSphere Application Server, being IBM's leading offering in the J2EE space, was selected as the application server platform for LMS 1.0. WebSphere has gone through a number of revisions in its product history. It seems to be continually improving and the versions up to the newest one have shown their ability to handle their load in very large deployments. I think one example that is commonly cited is eBay running on WebSphere.

When you have such a well established product like LearningSpace, which is in its fifth release, why re-architect it?

It's something that customers may not be aware of, but I think that software engineers are aware of: software architectures have a limited lifetime. If you look at desktop products, the office products that we were using in 1995 and the office products that we use today are radically different. Their internal architecture is radically different. Maybe they have, from an outside perspective, pretty much the same features. They all type, they all check spelling. But internally, they're vastly different.

And the same is true of server applications. The code base for LearningSpace 4 and 5 was initially created around 1996. Some of the things that we recognize today as best practices for Web application design, such as the use of the model view controller pattern or other design patterns, were not well-understood then. Those who have knowledge of the internal architecture of LearningSpace 4 and 5 understand that the code base is somewhat dated, and it limits the adaptability of the system.



One example of how LMS 1.0 differs from LearningSpace is the model view controller architecture that I mentioned. The presentation and rendering of all of the data in LMS 1.0 is completely separated from the processing or generation of the data. Aside from it being a good engineering practice, one thing that's good for customers is that they can produce entirely new user interfaces because of this separation. They only need to modify the JavaServer Pages that we ship with LMS 1.0. Another good example of how the architecture has improved is that internally in LMS 1.0, we have a highly componentized architecture. It's on the order of 100 to 120 internal services that the LMS 1.0 product uses to perform different tasks. So these services include directory manager service, enrollment service, and user lookup service. Partitioning our internal design into these separate services enables us to develop this system rather quickly. It's taken us about one year to bring the first version of this to market.

The new architecture also has a benefit to customers and systems integrators because they can easily replace any single service implementation with an implementation of their own. One example would be the service that we use to connect to a directory and to retrieve information about individual users. We support LDAP directories with our implementation and that includes the Domino Directory. If a system integrator wants to connect LMS 1.0 to an entirely different directory system, he can write a single, targeted component and plug that into the LMS deployment without replacing anything else, and it would just work. The previous LearningSpace 5 architecture could never accomplish that because it didn't have the internal component design.

Are you supplying APIs to allow customers to do this sort of customization?

Absolutely. Our public APIs are manifested as Web services. We learned from the LearningSpace 4 and 5 history that the main use of APIs really wasn't to customize a system at an internal level, but to connect the LearningSpace system to another application. Public Web services really are a much better way to do that than any proprietary API. We have Web services for things like searching the course catalog, enrolling a user, and provisioning a new user in the system. They use all of the Web services standards, including the WSDL—the Web Services Description Language—and any developer who has a compliant toolkit can access our Web services without knowing anything else about them.

There are internal APIs, as I mentioned, for creating these highly customized internal components and developers can utilize some of those. But, as I said before, I think the main use for that is going to be for system integrators or other parties who are creating a special version of the LMS 1.0 application for a particular environment.

Were you able to reuse any of your code from LearningSpace?

There was a lot of great code in LearningSpace 5, but the effort to take that great code and to recycle it into the style we needed for LMS took too much work. It was easier for us to start from scratch and to be informed by the lessons from LearningSpace 5 in the places where it did things well, but not to be bound by it.

Let's talk a little bit about the existing customer base and how customers can migrate to LMS 1.0 from LearningSpace 5.

There's a lot of detail in this topic. I wish I could say that it is a one-button, transparent migration, but it's not that

simple. One of the biggest issues to be accounted for in migration is our support in LMS 1.0 of LDAP as a directory. LearningSpace 4 and 5 really had no direct directory support. The list of users who were known by the application was maintained internally by the application. We had ways that you could import those users into LearningSpace—you can import them from Domino, for example—but if the user information changed outside of LearningSpace, the application would not know that. You would have to continually synch it back up. One of the things that we took as a requirement—and this is really an IBM/Lotus requirement for LMS 1.0—was to make an LDAP directory the sole repository for user identity and profile information. If your enterprise has an existing LDAP directory—and almost all enterprises do to manage email—you just connect that to LMS 1.0, and all of your user names and passwords that you use work automatically in LMS 1.0.

That's a big advantage for customers, but to go back to the migration story, we have to provide special tools that solve the different configurations of LearningSpace 5 or 4. Some customers put user names into LearningSpace 5 on an ad hoc basis. These users may exist in the corporate LDAP directory, but their user names have no relationship to what's in LearningSpace 5. We have some capability to perform LDAP searches on the corporate LDAP directory to help you map between the LearningSpace 5 fields and the new LDAP fields, but you may need to do some manual mapping to make that work.

For example, if your email address is stored in LearningSpace 5—email address is not a required field, but if it is in there—we'll use the email address to look you up in the corporate directory, and we will try to use that to map fields. We have strategies like that to try to figure out who you are in the target LDAP directory. But it's not something that we can do with 100 percent confidence.

Another area in which there are a lot of migration issues is content. Content is the great driver for learning system value, but is also the area that is the most complicated. We have a migration utility that will migrate all of the courses that you have in LearningSpace 4 or 5 into courses in LMS 1.0, but again there is some manual action that needs to be taken. One of the biggest design differences between LMS 1.0 and the previous LearningSpace product is we've added new capabilities and a new layer of organization in our catalog. In the LMS catalog, we have two different entities. We call one a course master. This is the template of the master copy of a course. The other we call a course offering. A course offering is a specific instance of a course—something that you can enroll in. We feel that our customers are going to find this a very valuable abstraction because you make the master one time, and then you create as many offerings of it as you like.

This is something that we originally had to do to support the classroom capabilities in LMS 1.0, but it's something that didn't exist in LearningSpace 5. There's only one abstraction in LearningSpace—the course. When we migrate, we migrate from the LearningSpace 5 courses into a combination of LMS 1.0 masters and offerings, but there's still some manual editing, if you will, of the final result that I think customers will have to do to get the content where they want it.

Another point there that I think shows what I mean is the LMS 1.0 catalogs. They support folders and a hierarchical organization. Again, that's something that didn't exist in LearningSpace 5. So one of the things that you have to decide when you migrate is which folders content goes into. Again, just to sum up, we have many, many more capabilities in LMS 1.0, and it's often difficult to map those new capabilities backwards to LearningSpace 5.

For those who have made a significant investment in LearningSpace, is it going to be difficult for them to migrate to LMS 1.0 because of the new WebSphere infrastructure? What plans do you have to encourage existing customers to migrate?

We are supplying WebSphere with LMS 1.0; it's bundled with the LMS product. You don't need to acquire that separately. The version of WebSphere that we include is the base WebSphere Application Server and a higher-end component that in WebSphere is called ND or Network Deployment. That is the WebSphere system component that allows the creation of clustered deployments. When users get this, they have essentially everything they need to configure the application server for virtually any capacity that they may want.

But there are a few other things that you need. You need your relational database, for example, but that's exactly the same as in LearningSpace 5. You needed your own relational database for that product too. You also need the LDAP directory that we've talked about, but for Domino users, they already have one. Domino can serve as an LDAP directory and publish its name and address book via LDAP.

Which relational databases are you currently testing with?

The platforms we test with are Oracle 8 and 9, DB2 7.2, and SQL server. This is really a customer-driven platform decision because these are database alternatives that are very popular in the marketplace, and they're the same ones that LearningSpace 5 supported. That's not really something that we can *not* support in our newer products.

What do you think is going to be the learning curve for system administrators who don't know WebSphere? Is a significant portion of their time having to administer a WebSphere server?

The changeover from the Microsoft platform to the WebSphere platform is going to take some effort in the area of education and new learning, to be sure. But I think customers are definitely going to find it worthwhile because we feel that WebSphere is a better performing platform that offers better management features. With the administration of the system, there are two main areas that you have to work with. There's the administration of WebSphere itself and that is done through the WebSphere administrative console, which is a Web application that allows you to do all the numerous or different functions you need to administer the server. You can install applications, start and stop applications, monitor applications, and define database connections, for example. All of those things, as I said, are done in WebSphere.

Then, there is another set of actions that are performed in the LMS application, and these are all actions that are very specific to the LMS operation. You can do things like configure the log files that the LMS application writes to—the level of log file output. Is it errors only? Is it warning messages and errors? Things like that. Then you can administer all of the users in the system, assign them roles and permissions, and other tasks of that nature. These are really application features, but I think our customers will see them as administrative tasks.

Let's talk a little bit about the developer and the end user experience. Is it going to change significantly for those audiences?

I think that the end user is not going to see much change, or if they do, it is all for the better. One thing that seems to have been a success in LearningSpace 5 was the improvements to the student interface, the interface that students see when they access courses and do all their tasks. We tried to carry forward much of those interface ideas into LMS 1.0. So, the tabbed interface of the home page, for example, is one of the features that carries forward. I think that users will like that our changes are more in line with well-understood Web application approaches. As I mentioned, the home page has a tabbed UI, but it also looks very portal-like in that there are separate fields on the page that have clearly understood uses. There is a portion of the page that shows you your active courses, another one that shows you your notifications, and another one that shows system announcements. We believe that any user, whether he is a LearningSpace 5 user or a totally new user, will find all of these elements familiar.

For developers, I think they will see a lot of differences. The APIs and all of the other points of developer interest in LMS 1.0 are much more Java and J2EE-centric than in the previous product. We talked about Web services—that's something that is totally new. It didn't exist in the previous product at all. The LearningSpace 4 and 5 event architecture, which was the main API that was accessible to developers, is also changing in LMS 1.0 because of JMS or Java Message Service, which is another Java standard feature. So the message there, just to sum up, is if you're familiar with J2EE standards, I think that you will be very familiar with what LMS 1.0 has to offer.

You talked a little bit about the limitations of the Microsoft platform. What do you think will be the benefits of using these new technologies in LMS 1.0?

There's a lot to talk about there. Let me start at a simple level. I think the first thing is that the Java componentry and Java systems in general are now so widespread that obtaining support or getting questions about Java answered is really very simple—JDBC driver issues, for example. If an issue comes up, you can search any of the well-known Web search engines like Google.com to find the answers almost immediately. It's amazing just how endemic Java and its integration technologies are in that way.

The approach of Java, I think, is another very big benefit. Essentially, there are no separate layers in the deployed application—it's all Java. In our previous applications, there were many different technologies that we had to bring together. Some of it was Java running in the Microsoft JVM, some of it was Microsoft COM components and ADO components, some of it was binary components that were supplied by the database vendors. These were all different layers that essentially the code jumped back and forth between when it performed basic operations. It could make resolution of a problem very difficult. Finding the layer in which the problem occurred was challenging.

In Java so much of that goes away. We can capture in a unified way a problem report that shows the trace of code all the way from WebSphere into LMS—into the database driver—and it becomes apparent where the problem is very quickly. Those are system-level issues, and it's difficult to see the benefit until you actually deploy a system, maintain it, and try to fix it when things go wrong.

Java has a significant lead over Microsoft technologies in the area of application integration. Virtually every large-scale system now has a Java API. There's nearly no problem with adding those APIs to the LMS environment. Whereas in other nonstandards-based architectures, if you have to get the API for your platform, you may not know whether or not the other system vendor supports the platform you're running on. The playing field is made even for all the platforms with Java, and that makes integrating all these different applications so much

easier.

Earlier you mentioned supporting deployments with upwards of a hundred thousand users or more. How scalability is LMS?

A big change that we have made in terms of scalability of the system is in the way we handle the session information for a given user. Something that is just a general principle when you build a Web application is that the more stateless it is—meaning you don't keep a lot of information about any specific connection—the more scalable it is. We have taken it as a design goal in LMS 1.0 to reduce this session level information to the absolute minimum that we need to support all the features. Some early tests that we performed show that even with the same hardware we can support anywhere from three to four times more users than we could with the previous application.

There's a significant cost savings in consolidation.

Absolutely. Where you may have needed three or four servers to support your user population, you may only need one. We also will have, I think, significant improvements in availability of the system which means the ability for the system to appear to the user as always on. In LearningSpace 4 and 5, there was a situation in which once your session was established on a given physical server, if that server experienced a problem or crashed, you lost all of your information. That limitation no longer exists in LMS 1.0. When you have a cluster configuration, one server of the cluster can go down, but users never notice any change.

With this change in scalability, do you see your target audience changing? Maybe a shift to the academic market?

I see the audience changing. An enterprise product that supports hundreds of thousands of users has a certain amount of overhead in its maintenance and installation that probably is too great for a workgroup to carry. I think that we're going to see fewer small deployments and an increase in the number of large deployments. That's just a natural change that I think is happening all across the industry. But that being said, I'm beginning to suspect that the answer may not be quite so cut and dry because the Java application server technology is becoming so widespread that it's all over the place.

You mentioned the academic market. Universities have students and graduate students who know everything there is to know about Java application servers. Some of them worked on the open source versions of the application servers. IBM often works with these organizations to seed them with licenses to WebSphere products so that they can utilize them or possibly return useful feedback to the product development groups. So, it wouldn't surprise me if the LMS product emerged in environments similar to that.

We also see interest for LMS in government deployments. They have the real potential to be the largest of all. A state government will have as many employees as a Fortune 1000 company easily. So that's another audience that is not purely corporate in its nature.

Do you have recommendations for capacity planning for this release?

Yes, that is something that our team is working on, so information will be available to customers. It will address server memory disk and CPU capacity and how many simultaneous users you can expect an LMS server to support. One aspect that we hope to address—and it's going to take a lot of detail work here—is providing the information in such a way that customers can put in how many courses they expect to have in their system, how many users are in the system, and how many courses each user will be enrolled in on average. There's probably on the order of ten or more variables that have a significant effect on the capacity planning answer. You almost need to create a spreadsheet to get the bottom line of three servers, four servers, or what have you.

I interviewed John Martin and Amy Travis from the LearningSpace - Virtual Classroom (LVC) team not too long ago. What do you think about having the Virtual Classroom component a separate product when it used to be a part of LearningSpace?

As an engineering task, there were many challenges there because they provided stand-alone capability to a product and integrated the product into another larger system. Essentially, the team was given two missions, and those two missions were not always consistent. The LVC team definitely deserves kudos for a good job there. I think it was, in a way, a kind of ground-breaking task because it's something that all of our systems are going to have to aim for, namely, this modular approach to product capabilities. With something like LVC, customers now have more options than they did previously. They can get LVC by itself, they can get LMS by itself, or they can get both together. When the two are together, they understand each other and work together productively. We've learned a lot as an engineering organization through the LVC effort, and I think it's going to help all of our products improve in this area.

Do you have any customers who have done something creative with the product, anything that may be

worth telling other customers about?

With LearningSpace 4 and 5, I have been involved in a few customer engagements, and I think the main thing that I have taken away from those is the creativeness in devising training or the things that customers apply training to. The straightforward view is that learning is for the well-understood—train how to use an office suite or train how to use an internal procedure. But many of our customers have used training for things that I never would have thought they would use training for, things like sale styles, for example. These are not hard skills, but the training is a useful way to communicate the information because it is structured and it makes people go through it in a more meticulous way than they would if they were just looking at a Web site.

One thing in the LMS that I think is going to aid that effort is the Authoring Tool that is part of the product. With LMS, we provide users the capability—or much stronger capability—to devise their own courses, and they can build those courses from different pieces of content that they import. They can take an existing course and customize it. Or they can make a whole new course from scratch. Then, once they have that course, it will provide all of the management and deployment benefits of LMS. All of us on the team think that we may see some interesting things in the authoring area of the LMS.

What lessons have you learned from those customer engagements involving LearningSpace 4 and 5 that you've put into LMS 1.0?

I think the number one thing is simplicity. Some of my early work on the LearningSpace team addressed critical situations in which the product was not operating the way it should, and it was difficult to find out why. In almost all of those cases, an important reason for the problem was excessive complexity in the code or excessive complexity in the configuration. In LMS, we tried to simplify the code as much as possible by making everything direct and independently testable.

Something that was not done in LearningSpace 4 and 5 that has been done in LMS is specific attention to what is known as unit testing. This is an engineering practice in which the internal components of the system are themselves independently tested outside of the complete learning product. We then have a much higher degree of confidence that our internal components are going to be correct having done that. That sums up the simplicity as something that leads to more reliability and greater serviceability for customers—that was the number one engineering lesson.

I think as a feature lesson, we learned to leave features as open as possible. You can never really predict every single need, obviously, and trying to address needs with a highly limited feature ultimately may be great for one customer, but it's going to disappoint another one. Because of our internal component structure, we can partner with a group like IBM Learning Services or IBM Global Services to create highly modified versions of LMS to suit a specific customer need. I think this need for adaptability was the other big lesson that I took away.

What are some of the plans for future releases of LMS that you can reveal to customers now?

The plans are to take this platform that we've created with 1.0 and to continue to add value to it. One of the big areas that we will be addressing is in content management. In LMS 1.0, we can maintain physical ownership of learning content. That allows us to do a lot of things that we never could have done in LearningSpace 5. We can automatically deploy content for you to one server or another. We can support servers that are geographically very separate, and that gives you scalability and other architecture options that you never had before. We intend, in our follow-up releases next year, to add even more features in the content management area that make us even stronger.

Another area is in enterprise application integration—integration with systems like accounting. This is an area where a very high degree of customization is needed. Large organizations have internal accounting systems, and they're virtually all different in one way or another. We have in our framework for LMS 1.0 all the necessary capabilities to do the integration with these different accounting systems, but now we need to actually start delivering on that next year.

The engineering team working on this has come a long way and built a lot of great things and is very interested in working with customers in actual deployments to understand the feature requests for the next release. We want to make the engineering organization as customer-responsive as we can and to get the engineers who actually work on the code exposed to the real uses of the product. The desire for enterprise systems is that they be highly adaptable and customizable. So we talk about, "Yes, we made our architecture adaptable," but until we actually get out there and see it working in the field, we will not be doing the best job that we can.

We are already beginning the planning for parallel engineering efforts. There's a certain amount of work that we understand fairly well that we could begin right now. There are other areas that definitely need more customer input before we can form a concrete plan. So, we will be working on all of those efforts in parallel. That will make for a different sort of pattern to the work than some engineers might be used to, but I think it's going to be

ultimately better for all of us.

ABOUT FERNANDO SALAZAR

Fernando Salazar has been developing software applications since 1985. He joined Lotus in 1995 to work on Notes Defense Messaging System. He later went to work on eSuite, where he was the architect for the eSuite Servlet Development Kit. Fernando joined the Lotus learning technology team in 2001 at the start of product and technical planning for the LMS project. Outside work, Fernando probably spends more time than he should developing and maintaining his wife's knitting and textile Web site www.wiseneedle.com.