VoicesinBUSINESS

CEP01 – Daniel Chait & Ross Hamilton, Lab49 (pt.1)

Announcer: Voices in Business presents the Complex Events Podcast, sponsored by BEA Systems. Show No. 1.

[intro music]

Announcer: Hello and welcome to the introductory episode of the Complex Events Podcast, brought to you by Voices in Business and sponsored by BEA Systems. In this inaugural episode, Daniel Chait, co-founder and Managing Director of Lab49, and Ross Hamilton, Lab49's Director of Client Engagements, speak with Vicki Zesses of Voices in Business, about the evolution of complex event processing.

In this first half of the two part interview, Daniel and Ross discuss the architectural changes in IT infrastructure, from batch processing to event driven, and look at how these changes were driven by the increasingly complex trading strategies that demanded both low latency and the management of ever increasing volumes of data.

They also discuss the critical issues to consider when reviewing complex event processing solutions.

Vicki Zesses: Welcome to our first podcast in the series on complex event processing, sponsored by BEA Systems. We have with us our co-host for the series, Ross Hamilton, as well as Daniel Chait, who is our resident expert and will be participating in the series. I will first hand it over to Ross to introduce himself.

Ross Hamilton: Yes, my name is Ross Hamilton. I am the Director of Client Engagements at Lab49. Daniel will be telling a little bit about Lab49, but I am responsible for a lot of our client projects that we do in capital markets. We have been working with a number of CEP vendors in the marketplace, as partners of Lab49, and we have also been implementing stuff for clients. Daniel, would you like to introduce yourself?

Daniel Chait: Sure. Hi, Daniel Chait here. I am the Managing Director and one of the co-founders of Lab49. Lab49 is a software consulting company. We build custom new applications within financial services.

I think, part of the reason why we were brought in to speak about this topic is that the kinds of applications that we are brought in to build for our customers center around areas where innovations in technology, or new challenges in the market cause them to see opportunities to invest in technology solutions that can enable competitive advantage within the market.

And CEP, in particular, is one of those key areas that we see a lot of interest, a lot of

adoption within the market. Everything is moving towards real time, and high speed, and event driven within the marketplace. And a lot of the systems that we are brought in to build for our customers - trading systems and pricing systems and the like - have been increasingly centered around CEP type of solutions within the past several years.

Vicki: And then, competitive advantage, you mentioned Daniel, that's the key way to upgrading, to changing.

Daniel: Yeah, I mean, there are a number of reasons why firms invest in technology. Sometimes it is cost savings, sometimes it is upkeep and maintenance. But, what we find are the areas, where we are brought in most often, I think, probably the unique capability of Lab49, is in those areas where the market is changing. So, there is new or more complex derivatives to be trading, or new areas of liquidity have opened up.

Firms need to build systems to take advantage of those new market opportunities. That's where the kind of competitive advantages comes into place. If you can price faster or more accurately than the guy down the street, well, you can uncover opportunities they can't. That's just a continuing ongoing cycle within the industry of driving technology adoption.

Vicki: And therefore, the reason for CEP.

Daniel: Absolutely.

Vicki: Excellent.

Ross: Why don't we talk about some of the architectural trends that we are seeing in the marketplace? I think that will help in framing the discussion around Complex Events Processing specifically. There have definitely been many changes in the marketplace over the last few years. And the market is becoming increasingly electronic and online, buy-side innovations from the hedge funds, pools of liquidity, regulatory changes, and of course, a lot of changes in the technology itself.

Lab49 has really seen these changes overall and revolutionized the market, and certainly Daniel and I have been talking about in the past what those changes have been - going from batch processing type systems to more of an event driven architecture that we are seeing now. So, maybe Daniel, you could provide a bit more color on that.

Daniel: Sure. If you look back at the way the IT and financial markets worked, say, a decade ago, it was very much driven around storing data and processing it in batch mode. So, a firm would process a number of trades throughout the day. They would record those typically in a database. At the end of the day, they would have a program that would run through that database and perform a number of calculations to see what their profit and loss was for the day, to calculate various aspects of their risk and exposure, etc.

Over the years, that mode of operation has become increasingly obsolete. So, initially firms wanted to do those same kinds of end of day batch calculations more frequently,

say, twice a day, during the day; to give them a more updated picture of where they stood and allow them to more accurately participate in the market, and adjust their strategies throughout the day.

So, they would run the same batch again, say, in the morning, with new market data from overnight markets. They would run the same mid-day as well. Eventually, that became a request to do it hourly, and then every 15 minutes, and before long, firms realized, 'hey, what we really need is real-time calculations for everything from derivatives pricing, risk or market data, P&L, you name it.'

So, that need for increased real-time visibility to view your actual position and various other calculations, drove the need for new software designs, and new architectures to handle them.

Ross: I think, one of the things that was interesting along that path was, you saw a lot of home-grown systems being created, and then a lot of point solution vendors came out with specific offerings, and then a trend towards industry standards and more platform-based systems. I think, that has brought us now to these much more open, flexible high-performance platforms that you can build all types of systems on.

Daniel: Yeah, definitely. In the classic sort of approach to one of these initial, early kind of real-time systems that we would see five years ago was to sit down in something like C++ or Java, and build an event driven system from scratch. So, listen for market data, and on every tick or on every so many ticks of that new market data, you would trigger some recalculation function downstream and send the data appropriately to wherever it needed to go.

And these systems worked, but they were incredibly difficult to build. They were relatively inflexible and they were not built on any kind of common application infrastructure or architecture.

Ross: We found that we were building lot of the plumbing, a lot of the infrastructure, as well as the business logic as well. And that really was not a scalable model then.

Daniel: You know, the analogy that I draw is to that of the applications server, or the Enterprise Java Beans. If you look back in through this early and mid nineties, you saw lots of people building applications from scratch to do things like e-commerce or content management-based web systems from scratch using something like Java or Coldfusion or a number of other technologies that were prevalent at the time.

Those applications, typically, had to handle the same kinds of infrastructures type of concerns over and over again, rewriting them from scratch each time - everything from web clustering to memory management to fault tolerance. What you found was that eventually this class of platform emerged, called application servers, which took care of a lot of those common considerations for you and let developers focus on the business logic.

So, CEP engines are really filling the same kind of niche in the event-driven or real-time application architecture space. So, it allows developers to write the business logic, and focus on how are you going to price the derivative, what your new risk is, and not worry about how do I connect up to market data, how do I distribute myself across multiple machines, how do I ensure performance and latency; those kinds of things are left best to the platform, where, I think, they are best served.

Ross: Right. I think, there have been a couple of specific trends that have really changed the market over the last couple of years. If you look at the hedge funds, the buy-side market, they have really been very innovative, they have really been leveraging technology to take advantage of these electronic market places. And that has really forced everyone in the market, particularly the sell side and brokerages, who having to service those guys, to support those types of businesses as well, right?

Daniel: Sure. It used to be the case that a hedge fund would download a spreadsheet or some other file at the end of the day that would indicate their positions, and they could look at that, again, for some sort of offline analysis. Increasingly, those types of firms are doing highly technical trading, doing high frequency trading, and they're requiring direct market access, or DMA, which is all leading those firms to a much tighter integration with their prime brokerages and with their other brokerage firms or broker dealers that they interact with.

So, all of that lends itself very succinctly towards an event driven type of architecture where, as things change on the market side, they're communicated directly to the client in a very real-time way.

Ross: Right. What about the adoption and formation of standards and open source? I think, there has been a pretty strong maturation process in terms of Java, and some of the other technologies as well; and we are certainly seeing a lot of open source technologies being deployed. And I think, that really helped build much more flexible, configurable systems. Things like the Spring Framework, for example, have definitely reached pretty reasonable depths of adoption.

Daniel: Absolutely. You know, it's an interesting time if you go back to the analogy I was giving earlier on application servers. Eventually, that market matured, but at the beginning there was an explosion or a profusion of different designs, different architectures, different ideas.

I think, you're seeing the same thing in CEP right now, so, if you look at what's out there, there are a number of different ways that various vendors have produced their systems, and some of which are more and some of which are less compatible with things like Spring Framework or other open source technologies.

So, it will be interesting to see, if you look at the evolution over the next several years, where that heads. But, certainly now, there are several ways that... certainly we've seen

things like Spring, and things like certain open source XML components and others interact.

Ross: In terms of implementing event driven architectures, what do you think are the core challenges organizations face?

Daniel: I think, one of the big challenges that I've seen is that from an application development standpoint, it's a different model. So, your developers are used to writing these declarative or so-called imperative programs - do this, do that, do the next thing. When you look at writing an event driven application, it's the opposite. It's kind of watch for this or watch for the other, and then respond in this way.

Ross: It's more data driven, right?

Daniel: It's more data driven, and so, it a bit of an inversion about how you actually think about designing applications. So, it does require some training and some ramp up time for your developers.

I think, a second big concern that we've seen from a number of different customers is around, how do these systems actually fit in with my overall architecture, or my overall IT infrastructure? So, what are the integration points, simple things like, what platforms does it run on? But, you also see, what market data sources and adaptors can I either buy or build for this platform; what messaging systems does it interact with, what's the high availability, and disaster recovery, and scalability...

Ross: How does it integrate with my existing system and existing proprietary analytics, right?

Daniel: Yeah. What methods do I have to integrate custom functionality that I may have written in Java, C++, .NET, what have you. How I can plug that all in, and not have to rewrite all that as well?

So, there are those challenges too, and I think, people are working their way towards different solutions to this.

Ross: So, let's talk about the actual CEP vendors and the solutions in the marketplace. You know, we've definitely seen the CEP market develop very quickly over the last couple of years, including funnily enough, you know, even a couple of years ago, we were building our own. And so, we were very excited to see a lot of these vendors come to bear in the marketplace.

Why don't you tell the audience Daniel a little bit about the core capabilities of CEP technologies; basically, how do they work?

Daniel: Mm-hmm. There are several different models for how CEP systems are built. Some of them are expressed as streaming SQL-like queries; some of them are expressed more as a rules engine, or some kind of event processing network type of arrangement.

But, in any event, they're all set up around this similar idea that you're taking the classic model of a database based processing, which was, you store the data, then you run processing over it. And we've inverted that model in CEP.

So, in CEP, what you do is you first express the kind of processing that you want to do. You phrase your queries, or you set up your event network, or your rules.

Ross: Or your scripts, right?

Daniel: Or your scripts.

Ross: Right.

Daniel: And then data flows through that, and those processing instructions fire in response to data flowing through it. Now separately, you may also write the data out to a database at the same time or somewhat later, but it's primarily not about storing the data, it's primarily about... as data comes in, we run it through these programs that we've written and process it.

So, that's the core of what CEP means. Now, additional to that, many of the vendors have included additional capability or components to that core CEP engine. They have server-side, and sometimes client-side, or developer tools as well.

So, on the server side, you see things for a high availability, disaster recovery, scalability, and clustering. These are systems that allow you to maintain your logic in the face of certain network outages, or hardware failures, etcetera, so that you can ensure to continue processing and handling the volume.

Some of the developer tools include all the kind of things that you would expect - ability to see your programs, debug them, step through them...

Ross: ... Using both standard tools like Visual Studio or Eclipse IDE...

Daniel: Sure.

Ross: ... Or even custom dashboards and development environments as well.

Daniel: Absolutely. If you look at where the market is now, there's still a little bit of flux around that. You've seen some vendors include or create additional Eclipse plug-ins recently. A number of them have created their own kind of studio applications, which mimic a lot of the functionality of those type of products; and others are looking at actually integrating with things like Visual Studio too.

So, there's still a little bit of flux there and that's one of those kind of considerations that, as a firm, as a customer, you need to bear in mind. What do I give my developers on their desktop to actually use these kinds of things right now?

Ross: Exactly. One of the key things about Complex Event Processing is the need for performance. So, particularly, nowadays, in the new modern-day trading rooms, you need to be able to guarantee low levels of latency and very high throughput, so, very high performance. And a lot of the vendors in the marketplace have definitely been talking about, "I can process a million messages per second."

One of the things that we always ask is, "Well, what are those messages and what are you doing with it?" Right? I think, it would be fair to say that if you don't have a high-throughput, low-latency solution, you're not even a player in the game. That's pretty much the qualification to entry in the marketplace.

I think, one thing I wanted to ask Daniel was, given that you've got some high-performance solution, what really are the things that people in the capital markets should be looking for when they're looking at these CEP solutions?

Daniel: Right, and that's a great question and it's particularly relevant now because there's a lack of benchmarking data and there's a lack of trustworthy information from neutral third parties, aside from a few small studies, about what are the real performance characteristics of these systems under real-world use cases and under actual market conditions and load?

So, what firms are doing, and I think, this is the right way to go, is that they're actually commissioning significant proof of concepts, where they'll go beyond downloading the thing and trying it out, and actually moving to specific proof points around handling this case of certain market data that we work with or looking data up from my-position database, and actually doing some real integration and work to get those proof of concepts built and see, for your particular use case, how does it perform in the real world?

There are a number of things that do affect that performance from, as you mentioned, Ross, from the sort of idealized messages per second number that a lot of them publish. So, these are things like, what's the model for calling out to your custom analytics?

A lot of the firms that we work with, they have research groups who spend their time building custom code, and sometimes it's C++, sometimes it's Java, sometimes it's mathematical modeling languages to do pricing or other calculations. What's the model for calling out to those calculations from within the CEP? Do you have go across a process boundary? Do you have to do some kind of dynamic code loading or reflection? Those things can slow you down. Versus is there a nice native high-speed hook to those types of applications?

Ross: The other key aspect of that is how flexible and how configurable is it, in order to be able to include those analytics or different analytics? How easy is it to deploy these new pricing models as they're being created?

Daniel: Agreed. Now, I'm going to add a second kind of performance characteristic, if you will, which I don't think everyone considers, but operating on a somewhat different and longer timescale here, what you have is, firms are actually competing against other firms in the marketplace. And if it takes you too long to build and deploy a system, well, then you've missed the opportunity.

And so, if the system you develop ultimately ends up being able to perform better in terms of number of messages per second, but it took you six months or nine months or a year longer to build it, well, you've actually missed out on a whole lot of opportunities.

Ross: I think, a good example there is where we are working with, you know, some of the buy side companies. They're really looking to get an entire automated trading system up and running within a couple of months and actually be trading as soon as possible. And they're actually doing that.

And we've certainly been working with our clients to architect solutions that are relatively flexible but get them up and running. And then, in three, four months, you can start to think about some of the broad architectural issues, like how do you feed your downstream systems and things like that.

Daniel: Absolutely. So, I think, the summary of that point is when firms are looking to buy a CEP or implement a CEP platform, one of the main considerations that they need to take into account is what's the development model and what's my kind of ability to quickly build applications, test them, deploy them, get them up and running? It's a separate consideration from runtime performance, but it's related in the sense that there's a time to market race that you're also participating in here.

Ross: Right. Right. One of the things that was quite interesting that we saw last year was when BEA entered the market. One of the things that was interesting to us in their capability was their deterministic garbage collection. And what that means is they're able to deterministically guarantee low levels of latency.

What that means is, for example, we've actually seen cases where traders have actually lost their entire day's P&L, because the underlying garbage collector has ran and paused the system and they've been picked off in the marketplace. We're talking about 250 milliseconds here, we're talking quarter of a second.

So, one of the interesting things that BEA came out with was being able to drop that upper ceiling to 10 milliseconds, and I know they're working on potentially five milliseconds. That's the type of thing that people need to be thinking about as they really scale out these solutions. How do you stop the actual architecture infrastructure affecting your ability to execute and be available in the marketplace?

Daniel: And again, it gets back to the question of how thorough is your analysis and your proof of concept before deciding on a platform, because things like network dropouts or garbage collections, they don't happen all the time, and so you need to make

sure that you're doing a really thorough job of testing that out under a number of different conditions and load scenarios so that you are sure to see all these kinds of cases and you're not caught when it's too late.

Announcer: That was the first half of an interview with Daniel Chait, co-founder and managing director of Lab49, and Ross Hamilton, the company's director of client engagements. In our next episode, we'll feature the second half of that interview.

For more information on Lab49 and BEA Systems, for details of how to subscribe to this podcast, or to leave comments and feedback, go along to our website at www.complexeventspodcast.com.

There's also links to Lab49, where you'll find the feature 'Inside the Lab,' containing some excellent demos, research and analysis, and a link to our sponsor, BEA Systems, where you can learn about their solutions for Complex Event Processing.

The Complex Events Podcast was brought to you today by Voices in Business, helping our sponsors achieve thought leadership in their business sector. For more details of what we offer, go along to www.voicesinbusiness.com. Thanks for listening. Goodbye.

[music]