



Extending an Ecosystem

Genuitec's president and cofounder discusses MyEclipse milestones and its feature set for developers working in an Eclipse-based environment

Interview by the Editors of Java Pro

August 28, 2006

Genuitec LLC prides itself on being the first to market with Eclipse-based solutions, and that distinction continued to hold in 2006 with the Milestone 2 (M2) release of MyEclipse 5.0 in the early summer—which coincided with the release of the Eclipse Callisto project—and then the more recent GA release of My Eclipse 5.0 in August. Genuitec's flagship MyEclipse Enterprise Workbench provides a feature-rich development environment that supports Java, Java Enterprise Edition (Java EE), and open source technologies.

The MyEclipse 5.0 feature list is extensive, but one noteworthy addition is the integration of the Matisse4Eclipse Swing User Interface (UI) Designer from the NetBeans IDE, giving developers the option to build rich graphical user interfaces (GUIs) in the same tooling environment available for building their applications.

Given Genuitec's solid association with the Eclipse Foundation as a founding member, a member of the foundation's board, an add-in provider member, and a founding member and sponsor of the Eclipse Plug-In Central developer resource site, *Java Pro* editors spoke recently with Maher Masri, president and cofounder of Genuitec, about the company's growing user base, the business model of its products and services, and his take on some of the current trends in the industry that are specific to the IDE space.

Java Pro: *Why don't we begin with your brief synopsis on the upcoming MyEclipse release announcement.*

Maher Masri: It started out for compatibility with Eclipse 3.2 and has full compatibility with Callisto. It will allow Eclipse 3.2 users to bring in any number of the projects that they need without imposing any requirements on them, projects that we already bundled: EMF, GEF, and others—WTP. The end user won't have to download those projects. The installer will be aware that they already have it on their desktop by virtue of MyEclipse being installed.

The GA release of 5.0 will bundle Eclipse as a full, certified stack, and it will be offered in two flavors. One is the download of the traditional MyEclipse set of features and the other is the download of the entire stack that includes Eclipse 3.2 and then the other supporting technologies



Maher Masri

that the customer would need to use. Also MyEclipse 5.0 extends Web services based on the XFire framework. (For more information on XFire, see the sidebar, "[MyEclipse's XFire Framework Bucks Tradition](#).")

JP: What is the business model for MyEclipse, and how does it differ from vendors that package together and test Eclipse components?

Masri: It's significantly different. MyEclipse is much more than a just a bundle of solutions. It's a fully integrated development environment that provides complete development tools out of the box without having to make end users try to figure out which open source solution is which. Most of the bundles out there take open source solutions and mash them up in such a way that installation becomes more convenient rather than the usability side.

Sixty percent of MyEclipse features are developed in house; 40 percent are implementations of open standards and open source solutions. It's a blending of the open source with in-house development and provides a seamless environment for the end user. Sixty percent of our features are not available in open source, unless... more than likely you'd have to pay for them independently. Our pricing model is based on a specific premise and that is our customers would be willing to pay for any one of our features at the price they pay for everything else on top of that.

The key to a subscription-based model is a value price base. It's the protection that the end user or customer finds enough value in the features that they're looking for. Research by Evans Data or Forrester basically concludes that specifically when it comes to this market, consumers are becoming much more value aware and much more component driven in the sense that they're looking specifically for what's within their limits [rather] than, say, an everything kind of solution.

Extending the Ecosystem (Continued)

And that's really the key question when it comes to MyEclipse as much as anything else, in that, as I said before our customers should be willing to pay for any one of the features that we offer, whether its UML tools, Oracle database development, Database Explorer, Spring, Hibernate integration, or any one of those capabilities. While it is a fully integrated, out-of-the-box experience, the user can turn on and turn off any one particular feature functionality and then use MyEclipse features in addition to other products out in the market as plug-ins as well as any competitive solutions that compete with us either on a stack basis or on a feature-to-feature basis.

Other companies out there are more motivated by selling services. They're more concerned with, and more focused on, training and services around Eclipse and around implementation of the client platform or J2EE product; whereas, we derive 99.9 percent of our revenue by the sale of the MyEclipse Enterprise Workbench.

JP: What indicators or observations led your organization to believe there was a market need for the type of integration Genuitec does on MyEclipse?

Masri: If you look at Eclipse beyond the IDE: that's something that took place as a conversation among the management team for Genuitec before we invested our resources to build the technology around Eclipse and before we dove very heavily into the Eclipse Enterprise Workbench. The analogy I want to draw is very simple. If you look at the early PC market it was dominated by a closed systems architecture that was at a certain price point. There were far superior technologies available, but then a company came in and commoditized the underlying architecture for that in an open source, if we should say, motherboard, for the PC, and that created in the market an opportunity that at the time nobody thought would be possible to create.

If you think of Eclipse in a similar light, if you think of Eclipse as a motherboard for application development regardless of the type of applications, then it's much more than just an IDE and much more than a tools platform. And that's why we saw in Eclipse in 2001, when we authored the first paper on rich client platforms, that it can be a lot more than just the tools arena that it plays into; and that gave rise to the RCP project when we authored that paper.

Why we looked at Eclipse specifically, why we did that then and continue to do so, is in one dimension, it's a very disruptive technology. In another dimension it's a large-scale ecosystem and a movement that has the potential to redefine an industry, and has done so very well. It continues to have the potential to redefine many more industries when you [consider] vertical industries.

I can think of every industry—and I've worked in almost all of them—that could benefit from having a common platform whereby solutions that talk to one another using a common implementation, desktop application, or even a framework spend 90 percent of their resources on integrating tools from different vendors and passing data back and forth between them. Eclipse in that context will solve that problem; it's just a matter of taking that context specifically to the vertical data set or knowledge domain and applying a common framework for that industry. It's going to happen. It's just a matter of time. We're incredibly better when it comes to Eclipse.

JP: Which, you touched upon when we met with you briefly at EclipseCon. Now that it's been a few months, do you still see that movement taking shape and moving forward?

Masri: It's a freight train. You're either on it or behind it, and it's one of those freight trains that some people can see it thundering toward them, and others don't even hear it until it's too late.

JP: Currently the MyEclipse user base is over 270,000 according to the MyEclipse Web site. What types of developers does MyEclipse target?

Masri: Let me put it in this kind of context: 50 percent of our users fall into the categories of one-to-six developers, whether they're working individually, in a consulting capacity, within a small group, or within a small company. That's one part of the population. The other 50 percent are your typical large-scale, large-enterprise, multibillion dollar companies that are building enterprise solutions. So our public is fairly heterogeneous, and our user base is as such.

What we have done to target the large population is make it so there is no friction whatsoever in the ability for our customers to acquire our software, to use it, to buy it, to continue using it, to provide information about it. We've made it such that that information is part of the daily habit, and the ability to find it is also a part of the daily habit. That explains why our marketing dollars are literally a fraction—a small, small, small fraction—of our competitors' marketing dollars, yet we attract more developers that are working with MyEclipse, using MyEclipse, than any of our competitors at the same time.

Extending the Ecosystem (Continued)

JP: At EclipseCon you mentioned that by the end of 2007 you're expecting to achieve a user base of 1 million. Are you still on track?

Masri: We're still tracked toward that [goal]. What the numbers on the Web site don't communicate is the actual true number behind the user base. We should try to figure out a way to do it, but the reality is that if one of our customers acquire, let's say, 1,000 licenses, or 500 licenses, from us, they count as 1 in that population—it's not 1,000, just 1. In that 270,000 that's just 1 for a customer who may have ordered 10, 20, 30, 60, 500, 1,000; that population, the 276 on the Web site is probably closer to 320,000 licensed users.

We're an incredible barometer when it comes to software and buying behavior. Think about this for just a second: if anyone is going to pay for software, and they're going to pay for a phenomenal set of tools that are offered, they're more than likely to pay first for MyEclipse than anything else. It's a no brainer, as our customers keep telling us.

JP: Given that 50-50 split between large-scale and smaller-scale development projects you just mentioned, do you anticipate the trend moving more toward larger enterprise-scale development?

Masri: No, surprisingly not. That composite has been stable for almost three years now. And that brings you two dimensions of thinking: Number one is that there's been a market that no one's serviced before. And if you look at our numbers, it's changed so much; it's not a small market. And the other dimension is that the size of the development group is shrinking, and we're seeing that even in large enterprises the team size is shrinking to a lower value, somewhere between five and nine.

JP: What are the most significant features that MyEclipse brings to the Java development community and why?

Masri: Do you have a couple of hours?

JP: Undeniably, the feature list is substantial, but maybe you can touch on what you think is most significant, maybe the top two or three?

Masri: The one that people keep talking about—one we've hardly marketed—is we offer probably the most comprehensive set of architecture server connectors for any one product. We also offer visibility, think on demand—an ability to talk to the application server in real time. Historically, users in building Java EE applications would have to package an entire project used in front of an abstraction to declare that project, start and stop that server, to assess the small delta change in one of the classes, in one of the syntaxes, or one of the UIs. And you can imagine doing that on a regular basis from a debugging standpoint. You can only do so many of those in a given day; whereas, MyEclipse, offering the think-on-demand ability, does it in real time.

You can literally change the syntax in the first or original editor, declare that solution abstraction before the delta change through the server, reinitialize the server in seconds, and be able to mix and debug the entire application from start to end in seconds rather than the many, many minutes, sometimes hours, as you're doing that in a regular development environment. Some of our customers tell us that that feature alone saves them two or three hours a day, and that's such a small feature when it comes to everything else that we offer.

The other capability is the Database Explorer (see [Figure 1](#)), and the more specialized Oracle database connector development. Many of our customers, large enterprises, have been able to save just on that feature a minimum of a thousand dollars a seat per developer for not having to go somewhere else to buy the same feature.

Extending the Ecosystem (Continued)

So the multiple features that are available, whether it's UML, or whether it's database development, whether it's J2EE development, or rich client development capabilities—again, I keep coming back to my main point—any one of those features alone would by itself satisfy the needs for the [same] price for everything else that is available.

Every time we release MyEclipse we talk about a small set of the features; naturally it would take several hours to talk through the feature set. I don't want to undermine that piece of it. [MyEclipse] 5.0 is a phenomenal release.

JP: When looking at the Eclipse-based IDE space, or even IDEs in general, are there other features outside of the MyEclipse feature set that you find impressive?

Masri: Every IDE has some capability that it competes [with] specifically. Obviously, there's an ideology when it comes to IntelliJ (IDEA) users. It is one that sets the tone for a lot of the customer base out there.

The remarkable, interesting technology that has the potential for success, we think, is what Adobe and Macromedia did with Flash in the Flex IDE. It's very interesting. Time will tell if it has the ability to compete with other rich client solutions, or desktop solutions, in large-scale implementations. As a concept it's very cool. Anything that has to do with an AJAX implementation, which is fairly interesting in the end, but seeing if there's anything there in the Eclipse space that is really even on par with what we offer, I really couldn't comment on that or who else is better there.

The key going forward is going to be in setting end-user trends and expectations [specific] to visual development. There's sort of a cowboy mentality that I don't need any tools that help me build solutions, and that will change over time as senior developers run circles around people that make those claims and make it such that they have to adopt more for sophisticated tools to build more complicated solutions. And there's really not a lot that we're seeing in the space where it comes to UI visualization yet.

JP: Given current trends in software development, is there added value for developers in some of the UI-building features too?

Masri: Yeah, sure, and again back to rich client development, we bundled Matisse for MyEclipse. We talked about [that capability] at EclipseCon. We took Matisse from NetBeans and integrated that into MyEclipse, and the features are available for Eclipse users. We tracked those kinds of integrated features on our feature list, and that gives us an understanding or a picture of what's going on in terms of the demand side out there. And it is there.

The demand for it though is not as near as the demand for the overall MyEclipse Enterprise Workbench, which kind of reaffirms the play that we've had when we initially conceived the product. Any of the features alone, if a company builds that set and tries to sell it by itself, the combinations of the sale for each of the different features would be far less than the sum of all of the parts.

For example, when we thought about building UML capabilities three years ago, we talked to a number of potential partners that we could look at, and we were awestruck by the number of licenses that were being sold in that space, which were somewhere between 100 and 200 licenses a month. That was like, "Wow! That's just a bad morning for us." When you look at what they're doing there, they're looking at \$2 million dollars revenue, selling \$1,000 software, and they're able to sell 100 and therefore keep their lights on accordingly. With \$30 software, you have to sell a hell of a lot more just to keep the lights on.

Extending the Ecosystem (Continued)

JP: Do you think that scripting languages are helping make Web applications development simpler?

Masri: We're going to see more blending of these concepts, even within Java and Java EE development. Enterprise ability already has a significant implementation of large-scale projects that are either legacy, or they are just ongoing and not going to abandon the whole idea as well as there are people building solutions very rapidly with PHP and AJAX, and others that have merit.

If you go to our site, our Web site is PHP based because we had to arrive at a simple conclusion when we first conceived of our chart idea: do we spend the next six months building the Web site, or do we spend the next six months building the product? It took half a person less than two months to put that power Web site together, and that's a fairly complex Web site from a back-end standpoint. I think every technology has merit. I don't expect that any one technology will become dominate in the near future. Dominance comes with critical mass.

JP: For the Eclipse-based solutions like RCP, does support make up Genuitec's consulting side of the business?

Masri: It did as we were building out the MyEclipse Enterprise Workbench. Today the consulting side is a very small aspect of our overall revenue. We're still courted by a number of companies, but obviously our top priority goes to servicing the 270,000 users worldwide first.

JP: Does that mean the consulting side will fade away gradually?

Masri: It's very selective. It's primarily done right now around Eclipse. That's the picture that we see that many people don't have an appreciation for. While you have 110 or so—130 actually—member companies in Eclipse Foundation that are building Eclipse-based solutions, there are literally thousands, if not tens of thousands, of companies that are probably building Eclipse-based, rich client implementations within their environments, and nobody's even talking about that.

JP: So wouldn't that translate to more of a need for consulting and for your expertise?

Masri: It would, but the market will grow to satisfy that if people need it. Again, it just depends on what kind of company you are. We used to be a consulting company; we have become a product company. There are a lot of companies building their businesses on giving away software to win consulting services. We're not that kind of company.

JP: Recently, Mike Milinkovich [executive director of the Eclipse Foundation] expressed some interesting opinions on standards, particularly in regard to stifled innovation being one cost of the JCP's otherwise successful efforts of standardizing Java. What is your opinion on standardization, particularly with respect to the issue of innovation?

Masri: Try to cross the street without a standard stop light. You have to have standards; it's the basis of social economy. You can't live without it. Now, standards can go so far as to stifle innovation, but you have to figure out which one is which. You can't just lump everything together, and I'm certain that that's not what Michael said when he talked about standards as they relate to innovation. It's dogmatic standards that are not pragmatic or don't have any merit that will indeed stifle innovation, and I think we are both on the same page on that.

JP: What's Genuitec's road map over the next three to five years?

Masri: Our customer base right now is in the development space, and we'll continue to support that indefinitely with the right feature capabilities. We'll continue to add more productivity features and more visual development tools. We expect that the combination of AJAX, Web services, and

enterprise service bus will mature to levels to be a fairly significant value set that will become standard. We expect customers will adopt more of the open source technology, for Hibernate, for Spring, for lightweight implementations on back-end solutions, and we're going to continue to support those technologies—also to just make it [possible for] someone with a minimal amount of experience to literally build a solution in minutes rather than having to understand a million abstractions to build a simple “Hello, world” [application].

This is my personal opinion, and it's shared by the company at large, Java and Java EE development have missed an opportunity to provide a real simple solution for the user community. People found out the hard way in the early generation of EJB development and related technologies that the reference implementation was just unduly and unnecessarily complex, and obviously the pendulum swung way too far to the left with regard to complexity.

Now it's swinging back to a more moderate point. Not everything has to be EJB, not everything has to be enterprise level and have 12 extraction layers; it's often necessary to build rapid applications to decide whether or not it's going to be successful and then harden it.

We adopted the principle, “make it work and then make it fast.” And we think that's the principle that a lot of the pragmatic community is going to adopt, and it's just a logical way to do it. We're going to continue to focus on building tools that will help to go rapidly to prototype solutions, and then take those prototype solutions into full production in a matter of a short time.