

Orchestrating Change

Fuego has an integration solution that is wildly different from traditional EAI solutions. Understanding the new approach — called business service orchestration — goes straight to the heart of the debate about traditional EAI paradigm vs. Web services.

To get the full picture, Tony M. Brown, editor-in-chief, eAI Journal, spoke to Fuego's founder, Felix Racca, and the CEO, Mark Theilken.

eAI Journal: What trends have you seen in the integration market over the last three years?

Mark Theilken: We see a lot of concern and skepticism in the concrete that's been poured using traditional EAI solutions. As a result we are seeing the advent of some serious interest in techniques like Web services to facilitate integration. We believe traditional approaches are too complex, inflexible, and way too expensive to implement.

eAIJ: Why do you believe traditional EAI solutions are too rigid and why a services-based approach provides the required flexibility?

Felix Racca: The problem with the data-centric approach is that it supposes there are going to be standards in data formats and ways to transmit data. If any of the current messaging buses were the de facto standard, or any of the IDLs or data formatting mechanisms used to exchange messages were the de facto standard, maybe it could have worked. But it didn't pan out that way. We've gone from spaghetti interfaces to spaghetti EAI. The big problem is that when you don't have standards, the only other possible approach is intelligence



Felix Racca



Mark Theilken

to interpret the disparate formats. This is why we believe a process-driven approach is much more adequate and flexible. Basically the key to this whole thing is that instead of creating adapters that generate a proprietary format, generate processes that drive services that can be in any format.

eAIJ: Are you getting feedback from the market that supports this view?

Theilken: Yes. Iowa Telecommunications Services made a decision to use a Fuego solution because they needed a completely integrated solution delivered in two months and we were able to do that at less than half the cost of using other technologies. I believe that the necessary adapters wouldn't even have been completed in two months, if they had used traditional EAI solution.

Tesoro Petroleum in San Antonio needed to improve SAP usability for their internal customers. For instance, their dockworkers in the marine services division require 17 screens and 80 clicks to create an order in SAP. Now Fuego automatically triggers the 15 back-end SAP transactions through the

SAP BAPIs. And they've also reduced their training time from three months to a few days. This is an example of the flexibility of our solution where we are actually taking this technology to front-end SAP and any other ERP solution to deliver a high level of improvement.

Racca: These are some examples of where we have been able to use a flexible business services orchestration-type of solution to help our customers focus on their customer-facing processes and on ERP improvement. In all of this, what these customers were looking for was reduced complexity, faster time-to-market, and the ability to rapidly change that solution as market requirements dictate.

eAIJ: What advantages does a distributed architecture give over the traditional hub-and-spoke?

Racca: Our paradigm is to use services that are exposed in the form of APIs through different technologies such as COM, CORBA, EJB, JMS, SQL, HTML, Web services, you name it. These are technologies, not applications. We can introspect the APIs that are generated on these technologies without programming. So this really accelerates the whole process, because we are invoking services from an intelligent orchestration engine instead of posting messages into a bus through an adapter that in turn invokes a service from an application. This is a huge difference in approach and basically it

reduces the time-to-market of any solution by about five times.

Theilken: And it can be deployed in a B2B scenario because this type of approach does not distinguish between a service from your back-end SAP system and a Web service from a business partner across the Internet. So B2Bi and EAI are absolutely the same thing for us. We are just creating a process that orchestrates services.

eAIJ: *So do you see yourselves in the Web services market?*

Racca: We see ourselves as leaders in the Web services market. To expose a business-relevant Web service, first you have to do your back-end integration.

Theilken: The B2B solutions of the future are going to require an easier way to integrate existing applications, as well as producing Web services. A Fuego-based solution actually is a Web service, and we look at Web services, as Felix suggested, as just another business service to be orchestrated.

eAIJ: *How exactly do you orchestrate business processes?*

Racca: Imagine that instead of having a Web server you have a process server, so process servers now can be totally distributed. I want to draw this picture in your mind of a Worldwide Web of process servers that run processes that actually invoke other services from other organizations or from back-end applications. The process server replaces the message bus.

Process orchestration server in Fuego is implemented basically as an engine written in 100 percent Java and provides connection pooling, thread management — pretty much like a very complex application server. It also provides clustering capabilities, fault tolerance, failover capabilities, automatic load balancing, and it interprets a process model that somebody designs using a graphical capability.

eAIJ: *So this process model would be deploying business rules?*

Theilken: Yes, you have the high-level graphical business rules, and then what we call our component integration lan-

guage, which basically has the ability to invoke services from disparate technologies as if they were a sub-routine of the process. You don't really care when you're building a process with this language whether it is a COM object, a CORBA object, or a Web service.

Racca: So with this engine, which really has the ability to drive services from anywhere and in any technology, you create a new compounded service that can be invoked by another process model. And this compounded service is exposed in the form of a Web service. This is basically what a process server is. You can't really relate to a process engine as a single entity because it is really a federation of process engines, just as Web servers are within the Worldwide Web.

eAIJ: *What are your thoughts on J2EE vs .NET?*

Racca: J2EE has the upper hand in terms of being platform neutral. Hopefully .NET will evolve to become platform neutral but it is really a way to expose service from a Microsoft platform. It is not a way to expose services from a mainframe or a Unix platform, or any other platform. So .NET is a way to invoke services that are built in a Microsoft environment. J2EE is more aggressive in terms of being platform neutral, precisely because it is written in Java, and Java was created with the idea of being platform neutral. We believe that for a process engine to be successful in being able to be cross-platform or platform-agnostic and create a Worldwide Web of services, it needs to be generated in J2EE rather than .NET.

eAIJ: *So you are saying that as we have a heterogeneous world, the multi-platform aspect of Java is what you need?*

Racca: Absolutely. I think it is something that needs to be said. However, bear in mind that true business services orchestration is not a platform. It must be platform-agnostic. So even though J2EE is the preferred way to construct this type of facility today, it will only continue to be the choice in as much as it is platform independent. A process server, or an orchestration engine as I prefer to call it, is not a plat-

form. It is more like a suspension bridge between platforms; therefore, it must be as open as possible.

eAIJ: *So would I be right in saying that the Fuego solution is a distributed, platform-neutral application?*

Racca: I think that's a perfect description. I would add: It's a distributed, platform-neutral app that runs process models to orchestrate business services.

eAIJ: *You mention reduced complexity and reduced speed-to-market as benefits of your approach but every EAI vendor I speak to says the same thing.*

Racca: If the other EAI vendors are saying they reduce complexity with respect to the point-to-point handmade connectors, I believe that. They eliminate the exponential nature of interface building. But that's all they do. And to do that, you have to build very complex adapters that talk to a proprietary technology, and then you have to add some complex logic into the process engines that usually run on top of that. These so-called engines are not an application that is invoking sub-routines like we are. They are something that reacts to events on a message bus.

You can see that the traditional approach really adds layers of complexity. You have the adapter layer that you have to build by hand, the bus management, the automator, or the process manager — all these things are totally different layers of abstraction and if you change, for example, an IDL on a message bus, you may have to change 35 adapters. So you have a domino effect. With us, none of this happens, because we have a facility that allows us to introspect the available APIs through any of the technologies that exposed them (COM, CORBA, etc.) and automatically generate an adapter. So there is no programming.

Theilken: In summary, what really makes us unique is, first, we have an ability to drive services from people, applications, and organizations, and do this transparently with the same engine. You're not going to find anybody else who does it. Second, our solutions are really driven top down by a process model that orchestrates the business services. For everyone else who talks about their process solution, their mantra basically is connect: build

connectors or adaptors, integrate, and then optimize. They start thinking about process when they are months into the project, rather than starting top down.

The third thing is that we have the capability to rapidly discover business services and bind them to the process using the automatic adapter generation technology that Felix talked about. And our solution is unique in that it provides the ability to deliver change on-the-fly — zero latency, no downtime.

eAIJ: *Is it true to say that your product inherently has technology adapters and then you create the application adapter yourself?*

Theilken: Exactly. It's created automatically by this feature that we call the introspective capability.

eAIJ: *And once I've deployed a Fuego solution to coordinate and orchestrate all these services, what am I actually ending up with?*

Racca: You are ending up with a composite application. You are ending up with an application that is exposed as a Web service. You create a process model with all the rules that invoke other services — maybe other process models or native APIs from back-up applications or people services — and that is exposed with an interface so that other processes can use it.

Theilken: The process model is an application — think of it as a supervisory application. And so it is the supervisory application that is basically running other supervisory applications — either it can be peer-to-peer or in a nested environment — and it is orchestrating the business services that we talked about.

eAIJ: *Let me get this straight. Everything is driven off the process model that orchestrates different services. This creates a composite application that in itself is callable, a service in its own right.*

Racca: You got it 100 percent. This model is iterative, totally distributed, and load-balanced.

eAIJ: *Why do you believe your model will replace hub-and-spoke approaches?*

Racca: Because the data-driven

model requires standards and you know that in XML alone we have 385 standards and counting. Whereas the intelligence-driven model allows us to create adapters that are totally reusable because you are invoking the service rather than receiving a message in a format that you impose. This is really a paradigm shift, and this paradigm shift is going to make B2Bi and EAI totally merge into one thing, with Web services being the major way to expose this one thing.

Theilken: This category convergence is really important and I think that we really need to stress how business services orchestration relates to EAI. I think it is also important that we explicitly define business services: They are what applications, people, and organizations do to fulfill a request from either an internal or external customer. From our perspective, these services are orchestrated into end-to-end business processes and it is important again that we make no distinction whether those services are internal or external to the company. The only difference in the implementation is if we use an XML/SOAP implementation so that we go through the firewall. And our customers have already done this, so they don't look at EAI and B2Bi as different technologies. They are not in the situation where they have to take one technology for use within the four walls, another technology for outside to connect to customers, and then add a process engine on top of it to integrate all those different technologies in order to deliver a solution. From our perspective, EAI is really a partial solution to the customer's business problem.

eAIJ: *How do you see the market shaping up over the next 18 months?*

Racca: Current EAI approaches are pretty much like the mainframe, and what we're doing is pretty much like highly distributed network computing. So I believe things are going to evolve in this way. Obviously, mainframes will continue to exist and so will the current approaches toward EAI, and we are probably going to be using these buses — because we can talk to messaging buses the same way we talk to CORBA, COM, etc. — to integrate more highly granular services within an orchestration. But I don't believe anybody in their right mind will use the current technolo-

gy going forward, as they see this new approach — the same way that I don't think anybody is going to buy a mainframe to do a new systems development.

So with respect to the future, I believe things will evolve toward the direction of Web services and composite Web services as we've described, rather than messaging and the traditional EDI or RosettaNet-type approaches also in the B2B space. I don't think that our competitors in EAI can evolve to a paradigm such as ours because the underpinning of what they have would require a complete restructuring for them to be able to come out with a product such as ours. I do believe that new companies will come out with things such as ours. I believe that the application server companies will try to evolve to something like this, but they will probably fail because again these engines need to be 100 percent platform independent.

Theilken: The important takeaway here is a quotation from Charles Darwin. Fuego, the name of our company, is derived from Tierra del Fuego, and when Charles Darwin first saw the island, he said, "A single glance at the landscape was sufficient to show me how wildly different it was from anything I had ever beheld." And that's us. We're wildly different from anything the market has seen before and we really think the demands of the marketplace require something wildly different, not just the same old stuff rehashed.

eAIJ: *Is there any danger in being wildly different? Will people see you as a strange paradigm and take comfort in the familiar?*

Theilken: That is without any doubt a risk but, as you know, these things will depend on how well we execute and how good the references are that we can come up with, and so far we have a perfect score.

We are not suggesting that this requires customers to make wildly revolutionary changes in their business model or their strategies to deliver solutions. What we are saying is that what's out there is totally inadequate for their business needs. They are pouring money and resources down black holes trying to make this stuff work. I think we can prove there is less risk and less disruption in delivering a solution with our approach than with using the stuff that folks have been using here over the past few years. 