LANINGHAM: This is an IBM Rational podcast. I'm Scott Laningham. Joining me is Stefano Sergi, the Rational Business Developer and EGL product line manager. He joins us to talk about building next generation business applications for System i. Stefano, welcome to the podcast.

SERGI: Thank you, Scott, my pleasure to be here.

LANINGHAM: Now, before we talk about IBM tooling for I5OS application development, please talk about the System i marketplace and even more specifically if you would, what kind of development challenges are these customers facing right now?

SERGI: Yes, well, you know, today there are millions of lines of RPG code and applications that run business critical function on the System i. And this [good] is really a tremendous asset because it captures years and years of business know how and basically runs the business.

But you know, as information technology evolves, companies are starting to look at new ways to use business information and to automate or streamline business process.

So as valuable as these applications are, and also as valuable as their developers who understand them are, they sometimes can get in the way of business innovation.

So what we hear from customers for examples is that they have problems with application documentation which either are very outdated or non existent, that they have difficulty bridging the gap between the emerging technologies and the available skill set, and also they're looking for tools that support the capabilities that their business users now expect.

So you know, a key decision that customers have to make is when and how to modernize these RPG or COBOL applications and how to provide more powerful and more functional ways for the business users to interact with information and the business processes...

And what is the best way for them to exploit these emerging technologies like Service-Oriented Architecture, Web 2.0, so that they can better integrate their value chains, their customers, their employees and so forth. Basically, how to transform that investment so that they can be, become more responsive to the business and become more competitive in the marketplace.

LANINGHAM: So it sounds like obviously modernization is a big area where these customers need help then.

SERGI: Absolutely, and let me elaborate on one of the

critical modernization challenge that I touched on, and that is the skill area, right. There are really different facets to this skill issue.

One is that the, what we call the skills mismatch. And that is typically the people who have the most knowledge about business domain are not the same who have the most knowledge about new programming models, new programming paradigms and new computing technology.

And vice versa. If you try to mix developers to deliver a full solution but these developers have different vernacular, different cultures, different processes and so forth, chances are that they will not work well together. And so the results are a system that can be somewhat fragile and difficult to manage.

So then another facet is that of the skills erosion. And you know, many of the System i veterans, the people that have developed applications in RPG for the past 20 or 30 years, you know, they're beginning to retire and it's difficult to attract younger developers to write business applications and business services for this platform.

You know, they expect modern tools, modern language, they expect to deliver systems and also they expect them to integrate well with the legacy system that's already there.

LANINGHAM: So it sounds like leveraging existing investment is something we talk a lot about as a chief point but also empowering IT staffs to develop new business applications. So talk about how they do that.

SERGI: Yes, well, it's not an easy task. There are different choices obviously. Basically the solution is to typically front end RPG with one of several different technologies that today support new computing models and new user interfaces.

For example, just to give you an example, you can write a whole new system and you can use Microsoft .NET or Java J2EE technologies and then use connectors that are made available by vendors for example to invoke RPG programs and exchange the data with the legacy systems.

Now, of course, this means that you have to gear up either C# or visual basic or Java skills which are all object oriented programming. So there are some alternatives, you know, there are now some server side scripting languages emerging like PHP, but they're also fairly complex, they're not easy to learn.

And then, this is not enough typically because then if you really want to create, you know, very responsive active, you

know, Web pages and then the ones that don't really require that you keep doing round trip into the server so that you can maybe process some logic for the user interface...

Then you typically, what you end up doing is writing some JavaScript on the browser side, right. So then you have to learn JavaScript as well. And by the way, despite the name JavaScript is very different from Java.

So it's a lot of steep learning. And, of course, because these are low level languages there has been an effort by the open source community and the vendors to develop what we call, what they are called framework that basically helps streamline the development experience in these particular languages.

For example, in the Java J2EE world you will find a number of different data access frameworks -- things such as Hibernate or Spring for those who are familiar with those. And you will find also user interface frameworks like JSF, Java Server Face.

So this is all good news, but however, as you adopt these things then your developers have to start understanding the architecture, the framework and the programming model that are the underpinning of the framework. And so there you go with another skill set that you have to add to this.

So bottom line is that developing these fancy modern end-to-end solutions, you know, typically requires expertise in either several programming languages and/or multiple people that specialize in these things. Customers are really looking for something to help them bridge this technology and the skills gap.

LANINGHAM: And now we're getting to why we're really here talking today aren't we? About what IBM is bringing to help out with this, let's talk about that.

SERGI: Okay, sure. As part of the i5OS V601 and the POWER systems announcement, we at Rational are announcing a new product offering called the Rational Developer for System i for SOA construction. Now that is a mouthful so we'll call it RDI SOA.

LANINGHAM: Okay.

SERGI: And this is an integrated development workbench that allows i5OS developers to do Web development and create Web services using their existing RPG or COBOL programs or service programs.

So RDI SOA is basically the integration of two products: the IBM Rational Business Developer and the IBM Rational Developer for System i. Now, the IBM Rational Business

Developer delivers the latest version of EGL which is the newest business language for the i5OS development community and it also deliver all the tools that are the IDEs for the EGL language.

EGL of course is a very modern language that, it's designed to shield programmers from those technical intricacies of Web development and SOA development. And with this release we're actually adding several enhancements that make it even easier to build modern business critical i5OS specific solutions.

For example, we added EGL verbs that allow you to access I5OS resources and data. We added the ability, for example, to use EGL to create portlets so that you can build portal solutions.

We added the ability to deploy EGL Web applications directly to the i5OS integrated application server so that you can exploit that lightweight integrated server. And we also have done a lot of work in simplifying and streamlining the ability to develop Web services using the EGL language.

LANINGHAM: Stefano, you mentioned EGL as the newest IBM business language. And I'm wondering where it is in terms of adoption in relation to other more popular languages?

SERGI: Right, well, you know, EGL has been designed from the ground up to support all the modern computer requirements and it was introduced in the marketplace only a couple of years ago. So it's certainly not yet as well known as some of the languages that I mentioned before.

However, EGL is in fact the combination of over 25 years of IBM research and development in the area of business oriented language like the Informix 4GL, like Visual Age Generator, which are currently used by thousands of customers worldwide.

Now, we're seeing it a tremendous interest in the marketplace on EGL and the EGL user community has been growing steadily so it won't be too long until it reaches a wider industry recognition, we are sure of that.

LANINGHAM: But how do you answer, say, the developer whose asking why would I want to adopt a proprietary IBM language like EGL?

SERGI: I'm really glad that you asked this question, and it does come up quite a bit. You know, all programming languages start out as proprietary by definition. They are the intellectual property of the person or the company that designs them.

Then the evolution of the language, you know, to a status or to a perception of non proprietary typically relates to either a process of industry standardization by a standards body or via a de facto broad based acceptance.

IBM designed EGL to be extensible and has every intention to position it as an open language so that we can allow third parties, you know, vendors, partners, customers themselves to use EGL extensibility model for their specific objective.

And we have already actually defined the specifications and we're in the process of socializing the specification with the OMG. In parallel, we're also pursuing the definition of a project that will be donated to the open source community. So of course IBM will continue to provide a commercial implementation of the open EGL specification.

So basically the answer to the question is that IBM Rational is very much aligned with the market dynamics and the trends that you mentioned. And in fact, you know, we have been and continue to be a powerful force behind open industry use of many of our core intellectual properties such as Eclipse and Jazz and now EGL.

LANINGHAM: Thank you for that. Now, and I'm also wondering where can customers find skilled EGL developers? Where do they look for them?

SERGI: Yes, well, you know there actually is an extensive network of consultants and IBM partners who have a lot of expertise and can help customers get their projects off the ground very quickly. And also, EGL is extremely easy to learn.

So in house developers can ramp up their skills in a matter of weeks, not months or years; weeks. And we even offer some Internet based free distance learning sessions to speed up that process. So it's really not a big issue.

LANINGHAM: That's great. Now, Stefano, we hear concerns about code generators that may be easier and quicker than hand coding and Java but they only deploy to a proprietary server. Is that the case with EGL?

SERGI: Well, no, not really. You know, applications generated with EGL can deploy as native J2EE or native i5OS programs exactly as you would deploy handwritten Java or handwritten COBOL or RPG applications.

There is a run time library that supports the execution of that code, right, of the generated code, but this is similar to some of the runtime library used by other compilers. But no, there are no special proprietary servers.

LANINGHAM: Okay, now what about questions of performance of the application it's generating, what about that?

SERGI: Well, you know, we have, we actually have hundreds of customers right now who are using EGL application in production and they have not experienced any perceptible difference between the performance of native applications and those generated with EGL.

We ran some internal informal tests, you know, for typical database business applications and we see comparable levels of performance, you know, although of course, you know, handcrafted code can always be tweaked to achieve real superior performance that you really need super experts and deliberate efforts to go do that.

But in general, you know, the typical business application, getting data out of databases, presenting it to the user for that cycle we have not really noticed any perceptible difference.

LANINGHAM: I'm wondering, Stefano, as a wrap up here, if you could talk a little about the type of i5OS applications someone can build with RDI SOA.

SERGI: Yes, well, you know, because of the combination of the products that make up RDI SOA -- as I mentioned, RDI

and the RBD -- you know, you can with RDI SOA you can build applications that actually meet any business requirement...

...all the way from BATS programs to online transactional system, you can build traditional RPG green screen applications, you can build Web services, you can create powerful Web applications, portal solutions, a real broad variety of solutions.

And we also have actually a tech preview of EGL that can help develop what they call rich net applications: basically, the Ajax made easy so that no only you can create very rich user interfaces but even do mashups with internal or external services. So the sky's kind of the limit here.

LANINGHAM: Yes, absolutely. Our guest has been Stefano Sergi, Rational Business Developer and EGL product line manager. Stefano, thanks so much for doing this.
SERGI: You're welcome, my pleasure. Thank you.

LANINGHAM: For more information on this topic go to ibm.com/rational/modernization. This has been an IBM Rational podcast. I'm Scott Laningham. Thanks for listening.
[END OF SEGMENT]