IBM

Moderator: Angelique Matheny
February 5, 2008
12:00 Pm CT

Operator: Good afternoon. My name is (Cassandra), and I will be your conference operator today.

At this time, I would like to welcome everyone to the conference call.

All lines have been placed on mute to prevent any background noise. After the speaker's remarks, there will be a question and answer session. If you would like to ask a question during this time, simply press start then the number 1 on your telephone keypad. If you would like to withdraw your question, press the pound key.

Thank you.

Ms. (Mathena), you may begin your conference.

I'm sorry, Matheny.

Angelique Matheny: Thank you, (Cassandra).

Hello everyone and welcome to this Rational Talks to You teleconference using the Rational Tool Set for Your Architectural Digs. I'm Angelique Matheny and I'll be your host for today's call.

Following up from Grady Booch's December Webcast titled, "Architectural Digs: Making Sense of Your Existing Application," today's discussion will focus on how you can leverage and reuse your specific existing infrastructure using architectural discovery with the IBM Rational Architecture Management Solution.

Our speaker, Celso Gonzalez, will provide expert insight and answer your questions about applying this concept and Rational solutions to your unique situation.

And as I mentioned, joining us from - leading this today's discussion is, Celso Gonzalez, Senior Software Developer, Model Analysis. He has 13 years of experience in software engineering including four years with IBM Rational. He's currently one of the Rational worldwide analysis design and construction leaders. His current role is to provide expertise to customers and field engineers and domains ranging from business modeling to J2EE development, including requirement management, architecture and design.

Now you won't find any slides for this conference. As I said, these calls are really for you. We want this to be interactive. And this is your chance to get your question answered directly and discuss what's on your mind.

And as the operator mentioned, you should press star-1 and she'll open up your line at any time.

Also, if you'd like to submit questions to our analysts after this teleconference, please email us at askusnow@us.ibm.com. That's A-S-K-U-S-N-O-W@U-S.I-B-M.com. Just put the title of this teleconference in the subject line and we'll make sure it gets to the right place.

Well, thank you for dialing up for me. So let's get started. I'm going to turn it over to you, Celso, and let you take it from here.

Celso Gonzalez: Thanks. Hi, everybody. It's Celso.

So as Angelique was saying that what we're trying to do is - after that (concept tool) approach that Grady was presenting last time trying to give you more insights and (income) how you can do that with our tooling, so that there is a - if you remember well, there is different - there is two steps on that approach.

So the first one is to make sense of what you are, what the assets that you are, understand what you are. Then, you need to get some attraction from those assets to be able to only record and only document the architectural aspect. So that's really what you're trying to do. You're not trying to get every detail of the code base that you have, but more the design - the main design decision, the main architectural points.

Then the other part of that information is you need to send that information somewhere because the purpose of that is not just to do that exercise for doing that exercise; it's to be about to reuse that information to make evolutions or integrate your solution without the solution.

And so that what we're trying to do here is show which tool on our tool set you could use and how you could use this then to do that.

So from a - and not only these - there's an - there is a different thing that you can do. So if your application is a Java application or a C++ application or set-up application, you can us the UML virtualization tool inside Rational Software Architect to get a sense of how that collaborates together, what are the different disconnected - what are the different sequences are going to be on my code. And then from that, trying to get a better understanding of the components and (how they relate). Then you can recall that information about the components and how they relate together.

If you're using more legacy applications like C, C++ or COBOL or you can use another tool that's called Rational Asset Analyzer. The Rational Asset Analyzer, what it allows you is to look at your code and you identify the relationship between elements, what are the main (piece), how they are related together. And then extract that information to help you make decision. What happens if I modify that piece of the code? What is the impact? What kind of modification will happen if I turn off that or if I connect that to something else? And that information will be done for you, so it will be gathered for you.

Another tool that will come to mind but still on the research aspect is called System (Broker). And this is developed by our (IE Bell Lab). And the idea is just simply trying to understand your application and trying to identify layers, components and that will opt you to make sense of the - of your role - architecture of the set-up application.

We are doing a lot of work also around pattern discovery. So if you remember, one of the things that Grady said last time was the code is the source of the information, but there is more information than that. There is things like which pattern did you apply when you developed that application or why the application is like that?

So some of that information is on document, on the mind of the architects or the design of that application. So that's something that you need to gather. But to gather that, there is no tool that will help you gather that information or get that information from their mind, so that you need to have inside views and then recall that information.

But for the pattern's aspect, there is tool. So we have an Architecture Discovery Feature in Rational Software Architect that can allow you to write in Java codes -- for 2 days only Java code -- to extract pattern information, which presently used and perhaps sometimes more important which entry panel do you use, so how (unintelligible) your application or where should you start your process to make less dependencies on some add components.

And that's where that overall tooling sense can help you. So it will help you to grab that information from the existing code that you have and you're going to have to use - going to all of those pieces of codes to try to figure out what is the relationship manually.

Then that information, you will be - you need to recall that information in a way that people will reuse it. So the idea is not to do that plan, then get that information to evolve new application to a new release and then drop that information. The idea is that if you don't pull that effort together, that information is to maintain that information. So you need to find a way to maintain that information for different tooling.

So then that means that that information that you gather, you need to make it accessible to your architect or your designer so that when they make a modification, add a new piece, it will make them a different factoring on the use in application, they can update that information so that the next team that will be coming knows what the architecture is about, how it’s working, how it’s evolved, and why some of those decisions have been made.

So that way the overall 808(unintelligible) mainly you and I are saying that there is too many tool that you’re using your Rational Software Architect. So the piece are around UML virtualization that can help you out, the piece about architectural discovery that can help you identify patterns and patterns on your application, as well as Rational Asset Analyzer that you have with your application, legacy application, COBOL, CICS application, et cetera.

So now, I’m happy to answer any question that you have about those tools or about what kind of information we need to recall, what should we recall, et cetera.

So, operator, please, open up the line for question.

Operator: At this time, if you would like to ask a question, please press star-1 on your telephone keypad.

We will pause for just a moment to compile the Q&A roster.

Angelique Matheny: Celso, we had a couple of questions come in early. Can we just start with just one or two of those?

Celso Gonzalez: Yup. Go ahead.

Angelique Matheny: And then while she - (Cassandra) compiles that.

What technologies - you mentioned just now, RSA and Rational Asset Analyzer. What technologies do these two help us?

Celso Gonzalez: RSA will help you with Java, C++ and C# application. So that will help you visualize that information.

For the pattern architecture today is only applicable to Java. But we are working on evolving that to broaden the scope to C++ and C#.

The Rational Asset Analyzer, this software is really to look at existing asset and analyze them and give you that outside view of what you get so you have a broader scope too. It goes from COBOL, DB2, IMS, CICS, and all the Java (SAT too) -- the J2E, HTML, (JC), Java - standalone Java application, as well as C++.

So those are the main technology that it will help you to gather information from.

Angelique Matheny: Okay, thank you.

(Cassandra), do we have any question?

Operator: There are no questions at this time.

Angelique Matheny: Okay. Then I will keep going with a few more.

Don’t be shy. These calls are for you. So just press star-1 and (Cassandra) will open up the line and you can get your questions directly answered by cell phone.

Okay. Next question that I have, it says when we recover the information, what do we do with it? And how do we record it?

Celso Gonzalez: So as I said, what is important when you gather that information is to make it reusable so that people can find that information and update the information even if they make modification. The thing is not to do it once and then for specific release and then drop it and ask to redo it the next time. So the idea is to try to maintain that information.

So what you need is to - what you need to do with that information is to recall that information on a way that people would be able to read and understand it. So that means you need to get that into document. It is the best way. (Do a mail) or any mean that people would be aware or able to understand.

Then you need also to make it available. So to make it available, the best way to usually to package that as a reusable asset, so it’s an asset that you want on asset repository. So we have such asset repository that’s for Rational Asset Manager that allows you to put that asset into Rational Asset Manager and make it available, and even add revision of that asset. So for example, you have that billing application that you have done a digging on it to try to understand what is the architecture. Then you get, let’s say architecture document about that billing application. Then you store it as a reusable asset into Rational Asset Manager, for example.

One interesting thing is that if you use Rational Asset Analyzer and Rational Asset Manager, there is an integration between them. So those assets that you track from Rational Asset Analyzer, you’re able to put them directly in Rational Asset Manager.

And what’s more important is that Rational Asset Manager allows you to access that information, but also allows you to add new version of that same asset. So that means that now, I’ve done that for (such) release. Now a new architect comes out; I need to do a new release of the application and we'll make some architecture modification or even some design decision. But make sure these are decisions that we would like to keep.

Then on that case, you can grab that asset, make a modification and send that asset back to the repository with a new version. That will let - everybody is now - is able now to trace also to which (relation) of the architecture for a given application is linked to.

And that's really the purpose for that work. So during that recovery, you can come back with a new (unintelligible) and code because you need to get the tools running, you need to take that information and then make sense and find a way to document and recall that information and on a document, on a model, on main model or mix between document and model.

And that makes sense only if you reuse that information and you try to maintain that information, because you don't like to, the next time that you have a new release to do - to go back to the same (effort) to rerun that information. It's better to be able to get that, to gather that because on the past, no documentation was done or nothing of documentation was done around the application.

One thing that is important to also know is that we're not speaking about getting every detail of the application. We're not doing a, let's say, (unintelligible) of the application into an IBM model that should use all the details. What we're trying to recall is really the architecture aspect, the main aspect, the thing that will help you to make the right decision if you're trying to evolve or to integrate that application with another one.

And that's really the information we're trying to grab. We're not trying to grab everything else, to recall everything. So - and another thing, if you'd like to make a modification, I mean that would be to an architectural level modification, so probably not everywhere you need to be making modification around the architecture of the application.

So that, once you have that, you would be pretty stable. But you need to make it available and make people the way of this existence so that if they have that kind of modification to do they make the modification on the asset and set a new version of the asset.

Angelique Matheny: Okay.

And, (Cassandra), any questions?

Operator: Again, ladies and gentlemen, if you would like to ask a question, please press star-1.

Your first question comes from (Mitchell).

(Mitchell): I was just interested if the asset analyzer does anything with power builder code?

Celso Gonzalez: Yes, I think it's about to apply to power builder code. So that means understanding the relationship and structure and get you that information. Yes.

(Mitchell): Thank you.

((Crosstalk))

Celso Gonzalez: Thanks.

Operator: There are no further questions.

Angelique Matheny: Okay. Here's my next question from the email box. Can we add our own pattern to the pattern detection tool? This is also my question, is, what is the pattern detection tool first?

Celso Gonzalez: So the pattern detection tool is a piece of Rational Software Architect. And that's called usually - when you see it on the tool it's called Architectural Discovery. So Architectural Discovery, what it does is look at existing codes and detect pattern that you have defined.

So there is a set-up pattern so that the (unintelligible) 1615 pattern that I was telling about on the tool when you get to Rational Software Architect. And there is a set of tool entry pattern, like if you go - that applies all those patterns that show too many dependencies on one or two elements on the application. That could be design of problem with the preferment the application.

So those are the one that comes from - with the tool, when you buy the tool.

Now you can because there's only an extensible framework, you can expand and you can create your own panel. Whether to date it's a little bit complex on the sense of trouble to write a lot of Java codes, to be able to extend and create your own pattern, whether it's a direction we are making of, making - one thing that we're trying to work on today is really trying to make that extensively easier. So getting you a way to create your own pattern on a faster way without too much coding so that you're able to create that pattern and then it'll be detected on a smaller amount of time.

Now if you're trying to do that today, it would be quite complex. You need to understand the (EPI), how it's works, et cetera. So our goal to this too on the software release is to make that easier.

Another thing that we're trying to do in (unintelligible) that has a link to that, is that pattern definition that we are using for the detection, what we're trying to do is to add a common pattern detection, so common pattern definition. Sorry.

What that means is that we have another tool in Rational Software Architect that's called Supply Pattern that allows you when you define a pattern together, to apply your pattern tool to a UML model by set of elements. Let's say that you decide to do the set-up pattern so that you have a pattern identified and then you're able to define which class plays which (walls) on my panel. And then that generate some more UML elements in relation to these things of UML element.

So what we're trying to do is to get those tools that are related to pattern to our dispensary and unique pattern definition. So that means that you create your pattern definition, then after that you're able to apply the pattern or to detect the pattern. And so in other time, what we're trying to do is also to make that creation of the definition way easier than it is today, whether the (unintelligible) is there. So today you would like to - if you'd like to create your own pattern, you can.

Angelique Matheny: Okay. I have one last question. It says do we need to use your (modeling), Unified Modeling Language?

Celso Gonzalez: That's a good question. Yes and no. When I say that - why I said yes and no, no because as I said, the main purpose of doing that is to have something that you can share with people. So if the people that will be using that don't understand UML, you need to go. So I would say you need to use a medium that would be understood by the people that would be then modifying or (mentioning) the application so they can understand the architecture.

So why I said yes, because I still think that UML is a good way to show visually the dependency and the relationship between the elements. So my - we're trying to say that if you can, I will encourage you to use UML. But if this will be too much gap, (skills) gap between the people that would be using the result of that digging and the people that are - and UML, then on that case, use another medium that the people are more familiar with.

Another option is to rather than using the tool UML is to use the subset of UML, or then to use the - what we call sometimes a Domain-Specific Language. So you can build your domain-specific language with context and entity that perhaps are more familiar with - for the people that would be using the results of the dig. And you can create a DSL using UML; for example using UML Profile so that people will not see the overall richness of the UML but just the subset with the concept that they're using on their day-to-day chart, so that would be way easier for them while (it's mentioning) for you the formality and the (semantics) of UML.

So that could be a medium term to help you with people that are not so skilled on the UML but keeping that visual UML kind of language.

Angelique Matheny: Okay. (Cassandra), any more questions?

Operator: There are no questions at this time.

Angelique Matheny: So if anybody thinks of another question perhaps after this teleconference, just make sure and email us at the askusnow@us.ibm.com mailbox and I'll send it directly to Celso.

Thank you very much, Celso. We really appreciate you sharing your knowledge and experience on today's topic.

Celso Gonzalez: Thanks, Angelique. Thanks to everybody who joined.

Angelique Matheny: To learn more about IBM Rational Architecture Management Solution discussed today, please visit developerWorks to get started.

If you would like to listen to this conference again or share it with your colleagues, this will be made for replay in MP3 format in about a week or so on the Rational Talk to You site, www.ibm.com/rational/talks.

Our previous teleconferences are available there as well.

I'd like to thank our guest, Celso Gonzalez for being with us to talk about Architectural Digs: Making Sense of your Existing Application. We appreciate you taking the time out to be with us.

We would also like to thank you, our audience, for your interest in IBM. We hope to see you back for another one of our events in the near future.

Thank you very much. Talk to you soon.

Operator: This concludes today's conference call. You may now disconnect.

Presenters, please hold the line.

END