

**Architectural Differences and Similarities Between Rules and Events**  
**Event ID: 346220**

Patty Brown: Good morning, good afternoon or good evening, depending on where you are in the world, and welcome to today's webcast, Architectural Differences and Similarities Between Rules and Events, brought to you by InformationWeek, IBM, and broadcast by United Business Media Limited. This is Part 4, the final installment of the IBM Good Decision webcast series, Architects and Developer's Crash Course in Decision Management.

I am Patty Brown, and I will be your moderator today. We have just a few announcements before we begin.

You can participate in the Q&A session by asking questions at any time during this webcast. Just type your question into the Ask a Question text area below the video window, and then click the Submit button.

At this time, we recommend you disable your pop-up blockers.

The slides will advance automatically throughout the event. You may also download a copy of the slides by clicking on the Download Slide button located below the presentation window.

And if you are experiencing any technical problems, please visit our Tech Webcast Help Guide by clicking on the Help link below the media player. In addition, you can contact our technical support help line, which is also located in the Webcast Help Guide.

Now on to the presentation, Architectural Differences and Similarities Between Rules and Events. Here to discuss today's topic with me is Jared Michalec, who is the Technical Sales Leader at the Worldwide BPM Technical Sales Group at IBM. He is a subject matter expert in IBM BPM as well as ILOG JRules. Jared is also IBM's foremost technical expert in WebSphere Business Events due to his previous role as the Field Technical Specialist at AptSoft Corporation, which was acquired by IBM in January of 2008.

Also with us today is Brett Stineman, a Senior Product Marketing Manager and IBM's WebSphere Product Group. Over the course of his career, he has been involved in marketing, product management and operations at various enterprise software companies, focusing on business rules management, business process management, content management and hosted applications.

So, welcome to both Jared and Brett, and I will now hand things over to Brett to begin the presentation.

Brett Stineman: Thanks a lot, Patty, and welcome, everyone.

As Patty said at the beginning, this is the last session of a four-part webinar series that was called a Crash Course in Decision Management for Architects and Developers. I hope that many of the audience today have been to our other three sessions. There's been some great content covered

by both people from within IBM as well as outside, a customer and an analyst that both have been talking about different concepts around decision management. And if you've missed any of those, every registered attendee can go back and view the archive of those various sessions, so I encourage you to do that.

But across all of the different sessions that we've heard, the first three plus what you'll be hearing today, we've really focused on this concept of some key technologies around what I would call operational agility and effectiveness. If we think of what our organizations do as being a business network, some of which is taking place inside of our four walls of our organization, back office transactions, interactions between different departments, but also we're all having interactions outside of our four walls with partners and with customers and with governmental organizations, etcetera. And if you think of that network that goes both within and outside of our organizations, we need technologies that will help us to deal with that, and especially with the decisions that need to be made on a day-to-day basis.

And there's really four key technologies that will help in dealing with this that we've talked about and that Jared will talk about some more as part of the demonstration that he's going to be giving. The first is around business event processing, and this is being able to take that data that is coming in from all of these different places across this business network and is being consumed by various systems at different points in time whenever there's a change in what we call state. So when something happens of significance within a system, we want to know what that is and be able to use that in such a way that we have an understanding of what's going on and are able to respond in a very timely and proactive manner.

Secondly, business rules management -- this is all about point-in-time decisions and being able to improve the quality of automated decisions that are taking place within critical business systems.

The third is around business process management because many of the activities that are taking place are within what I would call an orchestrated process. There's a series of steps that need to take place, hand-offs between different people, different departments, in order to get that process completed. And whether it be coming from event processing or from business rules, you want to be able to feed intelligent actions and responses into those processes so that the process knows what the next best step is, what's the right thing to do at a specific step in a process.

And the fourth is all of these technologies are generating data, and so you want to be able to capture that collected and utilize it in such a way that you can enable what we call continuous decision improvement. You want to be able to have snapshots of what's going on at a particular moment in time to dashboards and business intelligence reports, but you also want to be able to have technologies where you can get deep insight, data mining and predictive analytics and things like that. And you can use the results of those to improve what's going on within your business events, within your business rules and within your business processes.

So with that, I'd like to hand off to Jared, and he'll take it forward here and talk about the demonstration that you're going to be seeing. Jared?

Jared Michalec: Thanks, Brett, and welcome, everybody.

Before we go into the demonstration, I wanted to just walk through a quick slide illustrating some of the things you're about to see. The use case that we're going to talk about here is a fictitious automobile insurance company, and this is really meant to highlight the differences of different types of decisions that can be made by an organization, specifically focusing on -- surprise, surprise -- events and rules. And the reason that these are different is because you really see some more synchronous, more request reply, ask a question and get an answer type of rules in some processing as well as kind of a higher level view of what's happening in the overall enterprise, where the logic sits outside of the application as opposed to inside of it.

So we see on the slide here different customers coming into the website in order to apply for car insurance, and there are decisions being made -- whether they're eligible and what price we should show them. These rules are essentially sitting inside of the -- the logic is inside of the website application, and we can expose the changes to that logic to an external source, to a business rules management system, but the actual decisioning is happening in the flow of the application. It is requesting a price, the business rules system is returning a price, and we are getting an answer of prices, right?

At the same time, we have events happening along the way; these events are already happening. And whether or not we are capturing and evaluating those, that is what this new kind of technology is all about. Those events could be the start of an application, it could be the completion of the application; it might be the acceptance or the decline of an offer or of a quote. So these events are occurring over time, and we may want to start to watch for those patterns as they occur.

So on the business rules side, we're looking at more increased decision automation, ensuring consistency and also exposing these rules to the right user. The web admin is probably not the right person to be creating these pricing rules, and therefore we want to expose that out to the right person. And you'll see this in the demonstration.

And secondly, we have the business event processing capabilities, visibility across the channels. We may want to follow up on activity that includes the website but is not limited to the website, so multiple declines of a quote may lead us to want to follow up with a customer to figure out what's going on regardless of where that event originated from, whether it was the website or the call center or the branch.

So with that, I'm going to start the web demo, and at the end of it Brett's going to take you through a few more slides.

Hello, and welcome to a demonstration of WebSphere Decision Server, which is a combination of WebSphere Business Events for CEP or Complex Event Processing, as well as ILOG JRules for BRMS or Business Rules Management System.

In this scenario we want to walk through a sample website for a car insurance company in order to illustrate how these two pieces of the same product can be applied to different scenarios and

different situations as well as how they can work together in order to come up with better and more relevant business decisions. And finally, we want to talk about the differences in these two types of technologies and how you can apply one to one scenario and another to different scenarios as well.

As you can see on the screen, we have a sample car insurance website, so what we want to do is highlight business rule management first in order to illustrate how this synchronous business rule technology can be applied.

So as an end user, I'm going in as a potential customer to this car insurance, and I can start filling in my personal information. You can see that some of this information has already been populated. I can enter in some more details about this particular driver. In this case, it's Cindy Davis from New Jersey; she has a specific car, and there's some background information in terms of accidents and driver history.

In the next screen you can see I can enter in my car information, so in this case it's a 2006 Volvo S40. We can enter in additional information into this as well. So, so far, very basic information that you would find at any car insurance website.

The next screen I'm going to get a little more sophisticated and give very robust options in terms of coverage options and comprehensive collision options. So I can go in and choose different kinds of options here. And once I've made my decision, once I've decided on which options I want to apply, I can click Next. We can see that in the next screen I have several different pricing outcomes and a total price at the top per six-month period.

Now, the question is where did this website come up with all this information, right? And this kind of shows you the evolution of a business rule being implemented in a typical organization. So at the start we may have embedded some of this logic inside of the website itself. For basic information, that might be okay. So if we had certain levels of coverage that we were going to offer, if we wanted to just offer three tiers of coverage, then we could probably code some of that into the website itself.

Now as we get more sophisticated and more personalized, we can start to externalize some of that logic and maybe put it in a database or in a table or in a file in order to pull that information in in real time. But as companies become more sophisticated with their business decision logic, we really need to start looking at utilizing a business rules engine in order to get the most value of this information, because you can tell that all of this information, the person that is driving this information as well as where this information resides, is going to be outside of the website. And so we need a technology that's going to allow us access and allow us controls and governance in order to make those business decisions in real time.

So if I go over to my business rules management screen, I can log in and take a look at those rules that were applied for this eligibility and pricing decision. And from this screen, from what we call the rule team server, I can manage multiple rule applications at a single time. I can have governance and controls so only certain users see certain aspects.

So you can see I have different rules that I've applied here. In this case, I'm going to look at my pricing rules. Now on the left-hand side, I have numerous organizational features where I can put different rules in different folders. I can have different types of rules, business decisions for decision tables, rule flows, decision trees.

So, for example, if I look at my base premium rules and look at that comprehensive rule, I have a decision table that made that particular decision. So if I click on that, I'll see for this decision table, based on the vehicle value and the deductible, I'm going to have a series of different outcomes. So a decision table is really good at showing specifically, based on a certain combination of inputs, all the outputs that are possible. So this is much easier to look at than a series of if-then statements that you would see in a traditional business rule.

So if I go back to my website and look at what I chose here, I chose a \$1,000 deductible; I believe the car was worth \$40,000. So looking at my decision table, the vehicle value fell between \$30,000 and \$50,000, the deductible was \$1,000, and therefore my base premium should have been \$48. And looking back at that website, I can see the base premium for that car was \$48. So you can see all of the factors that fed into the total values are coming from these business rules that we've generated.

In addition, I could have other, more context-driven scenarios as well, so perhaps it's not just about the vehicle value and the deductible; it could be based on other factors as well -- number of accidents, the age of the car, the state of residence. So let's explore, if we're now the person setting up these business rules, if I decided I needed to add a new surcharge to a pricing table, for example.

So what I'm going to do is go back to my tree and go into my surcharges folder and, again, these are somewhat arbitrary, completely customizable organizational folders just like you would see in Windows Explorer, so we can really group our rules very logically and control them in that way. So what I'm going to do is add a new surcharge for New Jersey. We can see that there's already an SUV surcharge in place, but I'm going to add a global surcharge for anyone that's applying that is from the state of New Jersey. So I'm going to choose in this case to create an action rule, which is going to let us essentially write the rule in a very straightforward, English-sounding or English-looking sentence which will translate into a business rule in the execution side. So I'm going to call this new rule my New Jersey State Surcharge. I'm just going to set the status to deployable because I want to put it in right away, and we can see I'm presented with a template where I'm able to either type, so I can click and type, or I can choose from certain options, right? So you can see all the different conditions that have been configured for me for this particular applications; this is all done upfront ahead of where I am today by the IT user.

And what we're able to do is actually create an entire vocabulary or language around this particular rule application. And so we can really make this customizable for different types of assets, different types of objects, different types of conditions, and it can also be language-dependent, so we can have localization. So the rule that the person sitting next to me that's creating it may see it in a different language. So I might be seeing it in English; the person next to me might see it in Spanish, the person across from me will see it in French. Now, the very powerful thing about that is the underlying execution language is the same. So if someone next

to me wrote a rule in Spanish and saved it and then I opened it in my interface, I would see that rule in English. So that's a very powerful feature and, at the execution side, it's all in the same language; it's in the execution language. And so that's a very, extremely important piece that you don't always think about when you're going into developing a business rules application.

So let's go ahead and start this rule. In this case we were talking about adding a surcharge to New Jersey, so in this case I'm going to look at the state of residence and then I can enter a value, New Jersey. So there's my condition; now what is my action? What do I want to do with that? When I click on that, again, based on the vocabulary that we've set up ahead of time, I am presented with a list of options. And this is a contextual list so, depending on what I chose upfront, I'm going to get different values or different options when I click this list later.

So in this case, I'm going to add a surcharge -- and maybe we'll just make it a specific dollar amount -- to the coverage quote. So let's say that I'm going to add \$25 surcharge. I can also choose a discount in this case, but I'm going to add a surcharge to the quote and the reason is because it's a New Jersey state surcharge. And that's it. I have now developed a brand-new rule. You can see it reads very well. And now we're ready to save this, so I can go ahead and click Finish and now I'm ready to deploy it.

So depending on rules and responsibilities, depending on your authentication, certain users may be able to just change the state so that it can go into a review cycle. Other users may pick up that review and then approve it, and then move it to another state. And then other users will have the authority to go ahead and deploy that rule to the runtime when it's ready. But at any point, depending on what you're setting up, you can deploy these in a real-time environment. And you can actually have multiple rule versions running at the same time. So if I had some applications that needed a previous version of the rule, they could run against this previous version while I still have additional rules going in. And one of the most powerful features of the technology is the ability to control that governance, control that version control.

So in my case what I'm going to do is I'm going to go ahead and deploy this rule. And you can set -- you have different options. You can increment the rule version, so if I have a new, minor release of a business rule, some applications out there, like maybe the website application, is tied to always use the most recent version of the rule, whereas other applications, like maybe my BPM application, is using a very specific version of the rule so that it's not impacted by changes that are incremental along the course of the year. But then at some point you're ready to create a major version, and then your BPM app may point to that new major version later on after you deploy it. So in that case, we're just going to increment; we'll just replace the current one, because I'm not going to worry about versioning for this demonstration, and I'm going to go ahead and click deploy.

Now what it's doing is all the authentication, all the checks to make sure that all the assets are valid and that they pass all those tests, and then it's pushing that new rule application out there as a set of decision services that are callable by any application that has the authority to do so. And these rule decisions that we're creating or decision services, as we call them, are basically just like web services. So you can trigger those web services. There's auto discovery; there's WSDL that you can pull in. So it basically acts in your SOA layer in a very natural way.

So this tells me that this deployment succeeded, and now I'm ready to go back and maybe re-run my quote. So let's say that I go back through, and I'll re-start my quote process here. And I'll do it for the exact same person, right, so this person's going through, and I'm keeping the same options, same coverage options. Remember before it was per six-month period about \$350 for our coverage, and now notice it's gone up. It's gone up to about \$377. And why did that happen? Because of that New Jersey state surcharge.

So you can see these rules can take effect immediately, as soon as you're ready to deploy them and make them active. And, again, there's a whole set of checks and balances and governance that is happening behind the scenes to make sure that those rules are valid and that all the pieces are in place. So that's a quick coverage of business rules and how it can apply to this particular scenario.

Now all these, the business rules, as you saw here, are very much request reply. It's ask a question, get an answer; it's inserting logic into an application. In this case, in the website we are waiting to get to that next page until we get an answer back from that business rule management system, until we get an answer back from ILOG. And that is a very important feature of business rules because we are -- essentially that allows us to externalize business logic that already exists in custom applications. So BPM applications, job apps, websites, all of these can now externalize these important decisions that can now be managed separately from the application itself.

So in this case, the rules were being externalized because they really -- the web developer doesn't care what the pricing rules are going to be, right? So why should they be left to manage those pricing rules when you can externalize those and expose those to a pricing manager, to an insurance underwriter or anyone that really controls those factors, and they can make those changes independently of the website. The same applies to BPM or CRM applications, call centers, ERP systems, what have you.

However, there's another dimension here as well that we're going to touch on now which is asynchronous. Where is the sense and respond, the fire and forget? So along this way, we've just gone through it, and a simple example might be each time I create a quote, each time I request a quote, that is an event, right? And whether or not we're capturing it with an event management system or not, it's an event, right? The start of a quote is an event. The receiving of a quote is an event. Maybe not completing the quote is an event, the missing event. Choosing to decline or to accept that quote is an event. So you can see events are happening all throughout this application, and what we want to do is potentially capture those events in a very asynchronous, fire and forget way, so that it does not interrupt this particular application, but to coordinate and to correlate that activity and find patterns and act on those patterns at the right time.

So a simple example for our purposes in this scenario might be what happens if I apply for a quote and I get rejected, and then my girlfriend applies for a quote for the same car and she gets rejected, and then my mom applies for the quote for the same car and gets accepted?

Now independently taking each of those steps along the way seems like a very valid quote process, right? But that might be something that the insurance company wants to know about. They want to start to track that activity, not only by the person or by the application ID but also maybe by the vehicle identification number. So tracking these events across multiple dimensions and across time and then acting on it at the right time, deciding when to act, is what the other half of this product is all about, the business events piece.

So now we've switched over to the business events interface. I'm going to go ahead and log in here. You can see that this is actually being hosted inside of Business Space, which is IBM's Web Tool Mashup environment that allows for multiple interfaces and iWidgets to be controlled and customized in a single environment. So depending on which BPM and other technologies you have, you can actually mix and match and have different pages and customize these for all these different components.

In our case, we're going to take a look at the WB design interface, which is where we'll create these business event rules. So I'm going to open my project. Again, just like in ILOG JRules, you can manage multiple projects in a single environment. You can have access control in different versions of different projects all at the same time. In this case, we have a few different rules established inside of my events piece.

So you can see that this first rule is just looking at multiple quotes being started for a single customer, and there are a few different elements here that I want to point out. The first is the in response to, right, in response to an event. And that event is an object that has been defined for us that is a reasonable object or an abstract concept. The reason I stress that side of that, that object-oriented side, is because this object is really a truly reusable object in the truest sense of the word because at the IT level we can establish that these quotes are going to come in, right? No matter what the protocol is, whether it's web services or messaging or flat file or database pulling or however we want that event to come in, the end result is it will be this object, this BPM quote. We don't need to set the bindings or the rules or how that event is going to be used at the IT level. That's all set here at the business level. So we can use that and re-use that quote over and over again in many different situations without having to go back down to that integration level.

The second piece is the correlation or the filter, in this case, recent web quotes. So we can say if, under certain conditions, if there was a recent web quote -- and that simply means that the past occurrences of a new quote within the last week is greater than zero -- so, again, another reasonable component. And we're plugging these all together in order to create these event-driven rules.

And the last piece is the action. It's not just about finding and detecting these patterns. It's about doing the right thing when that happens. So in this case, we're going to push that action back to the BPM process or multiple processes. If there are multiple web quotes being started, we can push those all back down into the process. And so these events, these incoming events and these outgoing actions, can be any system you want. So in our case, in our demonstration, we'll focus more on the website, right, and receiving events from the website, correlating them and then



taking some action, sending an alert, sending an email, updating a database, updating a system, in order to do the right thing at the right time.

The last piece of this, which is one of the most critical pieces in the business events language, is this related by field. We call this the context. So when we say that there is a recent web quote, what does that mean? In our case, it's saying this is because it's for the same person. It's for the same full name. This could have been Social Security number, it could have been account number, it might have been, you know, a state, it might be branch, but that correlation ID or that context ID is being applied at the last step in the process.

And what that means is we can accept these events, and then look at them through multiple dimensions. So, for example, on this next rule, if we look at this, this rule is actually looking at events through the vehicle identification number. So it might be the same event -- in this case, it's a new quote coming from the website -- but these events that are coming in, we're now going to correlate them across vehicle identification number instead of by full name. So multiple web quotes may mean something different depending on the context that we are applying. So I can go ahead and take a look at what does multiple web quotes mean? It just means that the task occurrences are (inaudible) to. So this is the third time that this vehicle has been applied for a quote, and so regardless of the person actually doing it.

So before I deploy this and run it, I do want to show you how quickly you can create new rules as well. So, for example -- and you can see the template that shows up here -- in response to and I just choose any of the events that are available. Again, these events are established by IT, so they are set up as reasonable objects, but how they're used is completely dependent on the end user that's creating these rules. So if I say, based on a new quote, I'm just going to send an internal alert that says new quote, so very quickly I can pair these two objects together, this incoming event and this outgoing action. And if I were to deploy this, it would be running right away. So all of the heavy lifting has been done by IT in order to set up these re-usable components, but this level of abstraction allows us to create numerous rules along the way and be very iterative and very incremental in creating these assets.

The last piece of this is maybe I want to add a filter to this particular condition. So you can see I have several filters already created, but if I create a new one, I can create a new filter either based on data, either event data, or I can do it based on the correlation aspect, the number of times, the number of events that have happened or actions. So if I say the occurrences of an event or action, and maybe I'll choose if there was already an insurance new quote that has come in in the last two weeks, and if that is greater than zero, then that might represent that we've already seen a quote. And whether it's for the person or for the vehicle or for this region or for this type of vehicle or anything like that, that's all applied at the rule level. We're still at the filter level, so we can actually re-use filters in many different dimensions as well. I could re-use this filter, looking at it by vehicle or by branch or by insurance agent, so it's a very flexible platform and framework to create very sophisticated rules.

One other point here is also what if I said, rather than is greater than zero, it is zero. Now we're looking at the missing event, of we have not seen a web quote in the last two weeks. So this is another very powerful feature of business events, the ability to capture the missing event.

So let's go ahead and deploy this. I'm going to delete this rule because I don't really want this one to run right now, but I do want it run, that suspicious activity quote, for the same vehicle being applied, even if it's different people that are applying. So once I'm ready, once I've created all of my assets that I'm ready to push out, a very similar model to ILOG JRules, we're simply going to deploy or publish these assets out to our repository, to our runtime, and it's very quick. It's almost instantaneous to apply new changes to a project. So we've published this out, and now we're ready to go ahead and run the application.

Now since I don't have an investigations system up and running here, I do want to point out another few capabilities inside of the business space environment for business events, specifically around the testing interface. So if I didn't have the sample insurance website, I could actually generate my own events directly from the system and then track the progress and the outcomes of those events.

So I'm going to go ahead and go into my filters screen. And you can see I don't have any filters that have fired yet, that have executed yet, because I haven't sent any of events in. So now I'm going to go back to my insurance demo website, and let's go ahead and issue a new quote. We'll keep it for Cindy Davis, but we'll actually change this name later but we'll keep the vehicle the same in order to test this event processing application.

So the event will trigger when I've received a new quote, when I've gotten a new quote, so I've asked and requested for a new quote. If I go back to my business events application and refresh my filter screen, I'll see that I had a pattern evaluated where there are multiple web quotes; if we track this over to the right-hand side, we see it's false, right? So this is not true. If you remember, the rule was on the third web quote that's when we're going to issue that action; that's when we're going to trigger that alert.

So now let's go back to our insurance application. We'll start another application. In this case, we can change the driver information and maybe make it for John -- he's claimed three -- male -- we'll leave the other stuff the same. But you can see that I've changed some of the details. It's the same car, same coverage options, and now we've triggered another quote. Notice also this is slightly higher only because he's male, so just pointing back to some of those different rules inside of the business rules application tracking that.

So let's go back to our business space. We'll hit that refresh one more time. We'll see that second one that came in. It's still false, right, which is what we expected. So now let's go back and do it one more time, and this time it might be Cindy's mom. We'll leave, again, the other information the same, it's fine, issue a new quote. So that third time that came through, we're going to come back and hit refresh, and see that time it was true. So now we can go to our Actions tab and what actions got generated. It was start investigation with some details -- Cathy Davis, the vehicle is the same.

So, again, just showing and highlighting some of the capabilities here, the ability to watch these events over time. And it doesn't need to be, and usually is not, a single stream of events either. It's usually events happening across different applications, coordinating this together, correlating

them, and looking at them over potentially long periods of time -- it could be months or years that you're actually correlating this activity -- and then taking action at the right time.

So you can see that it's kind of the ying and yang of the decision management capabilities. You have the synchronous request reply from the business rules side; you have the asynchronous sense and respond capabilities in the business events side, and using them both together is a very powerful story.

Now, let's talk about when you would actually use them together in an execution standpoint. So it might be the fact that we've detected that there is potentially some activity occurring that we want to follow up on. In this case, the multiple quotes that got requested, but it could be anything. It might be an extremely quote request. It might be claims fraud. Whatever is happening that we may need additional information. We've detected when we need to do something. We've detected that, based on this information, based on this pattern of activity for this particular customer or for this particular vehicle, we know we need to do something, but we're not ready to start an investigation yet.

Instead of starting an investigation, I can change this to potentially execute a decision service. So now instead of sending it off to another application, now we're going to insert logic into our event processing application. So just like any other application, like a BPM system, website, ERP, whatever it is, we can actually insert business rules processing language into our event processing application in order to get a much more comprehensive view of what's happening or what we should do with this particular activity. And this where you start to see some synergy between the two technologies or the when to act and the what to do.

And so when we execute this decision service, we will receive an answer back, right? Ask a question, get an answer. We'll receive an answer back from ILOG and, based on that answer, we may do one of many things. We might trigger that investigation. We might trigger an internal alert. We might update a process. There are lots of different possible options that we could do, so I'm just entering in some data, details on that. But the idea is that you're leveraging the strengths of both of the components in this single product. So the ability to do sense and respond over time, over long periods of time -- sense when you need to do something -- and then triggering the decision service of what you need to do, what is the risk level, for example, of this particular activity.

One other aspect to that is the data that's being provided by the business events application, all this information that's happening on the event context, the past events, the previous information that's come through, all of that is being maintained in a very dynamic cache, in a distributed cache, and so all of the information is available for business events to make those decisions.

Now, in ILOG JRules, for the most part the information needs to be presented to JRules, to ILOG, upfront, so all the information to make a decision is presented in that request. So now using the two together, we're now able to present ILOG with contextual and historical information that's being persisted inside of business events so that it has much more information to work on and it can be much more targeted and much more valuable in its outcomes.

So that's my brief demonstration. I'm happy to answer any questions following this recording, and if you have additional information, please contact IBM. We're happy to provide customized demonstrations or discovery workshops as your needs permit.

Brett Stineman: Okay, so I hope -- this is Brett Stineman again -- I hope everybody found that to be informative, and Jared's going to come back here in just a second and talk a little bit more before we get into the Q&A.

What I did want to go over was some of the ideas that you saw here are very applicable across pretty much any industry. We used auto insurance underwriting, but the idea of situational awareness, being able to detect what's going on across various applications, various systems, various interactions, and then being able to then use that awareness to determine what's the right response in terms of how you want to interact with your customers and your partners or within your own organization is very powerful.

And so here you can see you've got multiple channels of communication that are going on with a customer, for instance, and being able to find those patterns that are taking place. And here we've got three different patterns set up so, one, customers making multiple requests and they're slightly changing one aspect of their request, so here they keep asking for different deductibles because they're trying to get different pricing -- well, that could mean that this customer is interested but they just don't quite -- they're not quite seeing what they want and so maybe you need to make some type of -- push some kind of personalized promotion or offer to that customer in real time to help get them past that and get them to make the purchase.

Another one would be customers coming back multiple times over time, maybe sometimes through a self-service channel, sometimes through a call center, being able to correlate those interactions to say okay, we need to get somebody involved, to call back the customer and talk them through what it is that they might want and figure out if we can come up with the best product. Maybe we need a person involved. You can't see that over on the right; it looks like there's maybe a little corruption in the slide. But anyway, it's basically saying how do we trigger a person to be involved and call back to assist the customer.

And then the third example is different things are being changed, multiple requests are being asked, and each one is slightly different, in such a way that maybe that's some kind of a potential fraud pattern that's detected. And then using the business rules capability to determine is it really a high risk of fraud and, if so, then what's the next best action to take? Do we alert a specific team? Do we let it go forward? Do we just stop the transactions? So there are a lot of different things that can come out of that business rule execution.

But in any case, I think the interesting part is hopefully that, no matter what industry you're in, you see the usefulness of this type of set of technologies around detection and decision.

I'm going to hand it back to Jared, and then we're going to go to our Q&A. Jared?

Jared Michalec: Great. Thank you, Brett.

So, yes, like Brett said, we just have a couple more wrap-up slides here. The slide that you see now, this is kind of a view of how we see the breakdown of the flow of different decisions that are being made, where we have the situational awareness, where we're looking kind of across the different systems and patterns and making asynchronous actions or other events to push back into the messaging layer.

And then we have more synchronous or more request reply, service-oriented decisions, so participating inside of the traditional SOA layer. So it's really similar to event-driven architectures and service-oriented architectures, where in most cases you need both and these types of decisions really play well in each of those different environments.

And then there are other side benefits that we see, such as monitoring, reporting, predictive analytics, which all play a part of decision management as a whole, but the ones we're focusing on today are really around those events and those business rules.

So event-driven architecture is really around being able to respond to these changes and these events at a higher level than just inside of the application logic. The service-oriented approach is basically you will link services together and make those calls, and you have to wait for the service to respond before continuing. And kind of the inverse of that is the event-driven approach, where we're sitting above the application layer, receiving events, correlating those events over time.

And then also there's another kind of sometimes missed requirement there which is event processing; it's not just business rules, event-driven business rules. But there is an element here that we need to have a running context and running state, and to have that extremely dynamic so that you can make these decisions not only based on the event that you just received but also based on other events and other activities that's occurred in the past. And so there's certainly another level there that we don't always talk about, which is kind of the runtime aspect.

But you can see the characteristics of the event-driven architecture here. It's typically one way, meaning kind of fire and forget; you simply omit the event and continue on with your day. You're looking at complex event processing, correlating events over time, the functional, the loose coupling, so you may have numerous systems involved in the event-driven approach, and systems may come in or drop out as time goes on, so very loose coupling, kind of a late binding capability. And you can see the other ones here but really, from our view, it's more the inverse of what we typically see as the service-oriented approach.

And as we said, you typically will need both sides of this coin in order to be successful. Certainly, in an inversion, one of our projects, you may -- you'll probably be able to get away with doing one or the other for tactical solutions, but as you grow into the more strategic, into the more enterprise-wide implementations, you certainly need both the event-driven as well as the service-oriented approach that you see here. That's really the main crux of this linking of these different types of decisions here at IBM, which is the business rule is linked with the event processing capabilities, which you saw in the demonstration there.

And then finally we have our wrap-up slide here. For next steps we have, as Brett said, this is Part 4 of the series, so feel free to log back in if you missed one of those to go ahead and download a recording. And also we are very happy to, as you can imagine, talk to you about this approach and these different types of solutions and what IBM can do to help you achieve your goals.

So at this time I think we'll turn it over to question and answer.

Patty Brown: Yes, thanks so much, Jared.

Before we head into today's Q&A, please fill out the feedback form that has opened on your computer. To complete the form, please press the Submit Answer button at the bottom of the page, and thank you in advance for filling this out. Your participation in the survey allows us to better serve you in the future, so we appreciate it.

So now let's move on to the Q&A portion of the event. And, as a reminder, to participate in the Q&A, just type your question into the text box located below the media player, and then click the Submit Question button. So let's see the questions.

Okay, Brett, may I ask you this one? Why wouldn't I just model decisions in a process workflow?

Brett Stineman: Okay. That's a very common question. People say okay, well, looking at business process management, why wouldn't I just all put it in there?

And the key two reasons are -- one is about re-use. So a specific decision service that I might use across many different functions within my organization, I want to be able to use that across different processes as well as other applications that might need it so, for instance, an eligibility determination. And if I try to model it into my process, then it becomes a siloed decision. And if I need to make a change to my criteria around eligibility determination, then I've got to go back into every single process where that is as opposed to having that as a decision service that's called out and make a change in one place and every different process that needs it or other applications just invokes the business rules system and it provides back the response. Change once, implement across many different places. So that's a key one.

The second one is around the complexity of decisioning. So if I have a high degree of variability from one transaction to the next, from one customer to the next, one geography to the next, and I have a frequency of change where what's true in the way I want to handle a decision today may not be true in a week or a month or ongoing, I need an easy way to implement both that variability from one transaction to the next as well as that change aspect. And the tooling that we talked about today helps organizations to do that in a very easy, safe and reliable way.

Patty Brown: Great. Thanks so much, Brett. Yes, thanks very much.

Okay, this next question I think can go to either Brett or Jared. How is the event detection engine configured to capture data from different systems?

Jared Michalec: I can take that.

At least in the IBM sense, our event detection capability follows a very unique architecture in that the events themselves are treated as abstract objects, meaning they're independent of the source that they came in on. And this is a little bit different than you would typically see in an integration scenario or other types of messaging, where you would maybe drag in a GMS or DHCP asset into the palette and then call that an address change.

For example, in the IBM technology it's the opposite. You would create an address change asset that can be applied to any number of different event patterns, and then you can bind that asset to a connection protocol. And whether it's one of the out-of-the box connectivity strategies, such as all the thinner protocols, GMS, DHCP, web services, database, the events can come in through any source. And it can be active or passive, so we can actively pull for changes or actively receive or passively receive messages or web service calls. And it can also come from any number of different native integration, from other IBM technologies such as CICS, IMS, IBM BPM, IBM Business Monitor. So there's several IBM technologies that can also natively admit events directly to our eventing platform as well.

Patty Brown: Okay, perfect. All right, Jared, you have one more question, and I think we have time for about one more question, so let me ask this.

What are the most common communication patterns for business rules and business events execution?

Jared Michalec: I would say -- I'm thinking a communication pattern here meaning the way that we receive events -- it's typically over some sort of messaging protocol, so either through an ESB or through MQ, for example, where the events come in in that way.

And then for business rules, it's typically through the web service layer. It's treated as essentially a web service call, what we call a decision service, that's able to execute that decision in real time.

Patty Brown: Okay, perfect. Well, let me just ask one more question then. This just came in from Maryanne. What is the underlying chakra behind the scenes? Is the IBM FileNet P8 used in this demo?

Brett Stineman: In the demo we showed today, it was specifically two WebSphere products, two products from the WebSphere brand. One is ILOG JRules for the BRMS, and then WebSphere Business Events, which is our CEP capability. And both of those technologies make up a single product that we call WebSphere Decision Server, which is the core of the demo today. And then there was a website, but that was just built for demo purposes.

Patty Brown: Okay. Well, great. Thank you so much. And if there are questions that were not addressed, I'm sure that we will follow up with you.

So, Jared and Brett, thank you very much. We really appreciate it.

For more information related to today's webcast, please visit any of the resource links that will open before you, and within the next 24 hours you will receive a personalized follow-up email with details and a link to today's production on demand. Additionally, you can view today's event on demand by visiting [www.netseminar.com](http://www.netseminar.com).

And don't forget, if you missed any of the webcasts in the IBM Good Decision webcast series, Architects and Developers Crash Course in Decision Management, you can also view these on demand at [InformationWeek.com](http://InformationWeek.com).

Thank you for attending today's webcast, Architectural Differences and Similarities Between Rules and Events, brought to you by InformationWeek and IBM. This webcast is copyright 2011 by United Business Media LLC. The presentation materials are owned by or copyrighted, if that is the case, by InformationWeek and IBM, who are solely responsible for its contents, and the individual speakers are solely responsible for their content and their opinions.

On behalf of our guests, Jared Michalec and Brett Stineman, this is Patty Brown. Thank you so much, and have a great day.