

JRules v7 Testing and Simulation Demo Audio Script

Introduction

Welcome to this IBM WebSphere ILOG JRules demonstration

This video will use a real-world scenario to show you the strengths of the WebSphere ILOG JRules Business Rules Management System, also known as the JRules BRMS. In this demonstration, you will see how it allows organizations to easily adapt business rules while ensuring compliance with policy requirements and analyzing the business impact of changes against key performance indicators.

This demonstration is based on a fictional car insurance company whose aim is to provide a highly customized car insurance quotation online from its website.

You will see how the JRules BRMS is used in the back office to create eligibility rules that will allow the web application to automatically screen auto insurance applicants. The eligibility rules will be managed through the following products and components:

The WebSphere ILOG Rule Team Server business user web interface from which rules can be managed.

The WebSphere ILOG BRMS rule repository where rule-based business policies are stored in order to be shared and re-used.

And WebSphere ILOG Decision Validation Services which provides testing, simulation and tracing capabilities to the various user environments in the JRules BRMS.

Adam, a policy manager, will log into the Rule Team Server to access the Eligibility rule project.

From this project, Adam can access all the rules that define the automation of business decisions based on the company's eligibility policies. These rules will be used to accept or reject an applicant for car insurance depending on the applicant's profile and driving history.

As an example the "Too Many Accidents" rule will classify the driver being a high risk if he had more than 4 accidents in the past 5 years.

Rules can also be entered using decision tables as shown here. In this table the driver will be classified as high risk depending on the combination of the number of tickets and number of accidents he has had over a defined time period.

Now Adam will apply the rules against some pre-defined tests in order to ensure that they are compliant with the corporate business policies.

A new test suite is created using Decision Validation Services, or DVS, from within Rule Team Server. The test suite is identified with a name and can also be documented.

Now Adam must decide what rules must be tested from the project. By default, all the rules from the eligibility project will be used for the test.

The next step is to provide the scenarios for the test. A scenario provides input data and expected results for these inputs. Scenarios can be defined in a Microsoft Excel spreadsheet.

Every test suite can also be versioned automatically.

The test suite is now created and ready to be executed. Let's have a look at the test scenarios file.

Test data in Microsoft Excel

Each class of information used by the rules to automate decisions can be populated with input data in the spreadsheet. In this example, 11 test scenarios have been defined.

For each scenario, input data are provided such as driver data, driving history, the various coverage requests that can be chosen, and the vehicle usage describing the percentage of car usage for each driver declared. As can be seen here in the first scenario, 2 drivers are sharing the same car.

The last input data is the vehicle page, providing details such as safety features and vehicle-specific information.

After the input data, the expected results tab describes the eligibility results for each scenario. Here only the eligibility status is tested. If a result of a rule execution does not match the expected result of a scenario, an error will be shown in the test report.

Additional test information, such as an expected rule execution time or list of rules fired can be added as test parameters for every scenario of the test suite.

Running a test suite

Adam will now execute this test suite using these 11 scenarios on all the eligibility rules of the current project.

The execution is now finished and the report shows that all the scenarios are successfully corresponding to the specified tests. Some execution details are displayed for each scenario so that the user can see what has happened during the test.

As an example, the first test scenario is successful. The eligibility response complies with the expected result as well as the execution time. We can also see the list of rules fired in this scenario.

The eligibility rules comply with all tests specified in the test suite.

After ensuring the rules are compliant with the business policies, a simulation is run to measure the effect of the eligibility rules on the business. A new simulation is created in Rule Team Server. A scenario service provider allows the company to measure the results of business rules against key performance indicators, or KPIs, using historical data over a specified time period.

The simulation is processing using the historical data stored in a database. In the background, all the eligibility rules are being executed using this input data to compute against the defined KPIs.

After several minutes, the simulation report is ready. This reports shows the simulation results and the computed KPIs . Over the 400 cases, we can see that 18% have been rejected during the eligibility process. These results can also be viewed for each state where the company operates for a more detailed analysis.

The company's auto policy business team has seen that too many young drivers have not taken a driver's education course and are involved in a disproportionate number of accidents, creating excessive losses to the organization.

A new eligibility rule must be added to filter certain individuals and to reject them during the automated eligibility review.

A new rule is created in the high risk driver package using the Rule Team Server interface. Adam selects a predefined rejection template to start his rule creation.

After entering some rule properties, the rule can be defined using a point-and-click editor. At any stage of the rule creation, possible rule definition text is displayed to assist the user. Thus the rule can be created without requiring any technical skills and in an easily understandable syntax.

After entering the rejection message that will be displayed in the web portal interface, the rule creation is finished.

The rule is stored in the repository – it is not yet deployed but can now be tested for correctness and simulated to measure its impact on the business.

The first step is to ensure that this new rule complies with the corporate eligibility policies. Adam will do this by running the test suite again.

After its execution over the 11 scenarios, a test warning appears clearly showing that a scenario failed. Some expected results are not matching the test execution due to this new rule.

The test shows that the "High Risk Students" rule has been executed, rejecting the request whereas the test scenario requires this set of inputs to be eligible. At this point, Adam or one of the company's business analysts must look at both the test suite and the rule to determine whether the new rule meets the business policies – in which case the test scenario needs to be revised – or whether the test scenario is correct and the rule needs to be revised.

After assessing the new rule and the test scenario against the business policies, it is determined that the new rule is too restrictive and must be revised.

Adam uses the quick edit mode to perform this rules change without having to go through the entire set of rule edit menus. The number of allowed accidents is increased and the rule changed is saved.

The test suite is executed again to check that the revision to this rule complies with the test scenarios.

The report shows that all the tests have been successful.

Adam can now upgrade the status of the rule from "new" to "defined", showing that it can be reviewed by other members of the policy management team.

Adam now measures the business impact of this new rule against the associated KPIs.

The simulation is going to be performed using this new eligibility rule – Adam can run the same data from the previous simulation.

After several minutes, the report is ready and the KPIs can be displayed.

The business impact of the new rule can be measured against the KPIs. The rejection rate has increased in a reasonable way, showing that the new rule meets the company's business objective.

Impact analysis

Using a Rule Team Server customization, Adam can compare two simulation reports and display the KPIs side-by-side. The comparison between the KPIs before and after the creation of the new eligibility rule clearly shows the impact of the change before deploying the new rule to the production system.

Since this change meets expectations, the rule status can now be changed to "deployable". During the next deployment to the JRules Rule Execution Server by the IT Operations team, this rule will take effect in the production environment.

Conclusion

As you have seen in this demonstration, the JRules BRMS provides valuable benefits in the management and execution of decisions in business systems by:

- Providing an easy, safe and predictable way to implement changes
- Reducing the time and cost of managing changes to decisions that guide business systems
- Allowing IT and business to collaborate in the rule maintenance life cycle, by providing the right tool for each stakeholder
- And increasing the visibility and understanding of the decision logic that drives system behavior

With the JRules BRMS, you have a comprehensive set of capabilities to manage rules with confidence.

Should you want to get more information on JRules or other WebSphere ILOG BRMS solutions, and to download a free trial version of our BRMS products, please visit our website.

Thank you for your time in viewing this demonstration.