Clifford Black
David Carlin
Nick Faccenda
Jacob Kershaw
Bryan Nunez
Damen Tomassi

Rule Framework Decision

The first requirement stated by ASRC for this Rules Based Decision Framework Aid was to "evaluate more than one open source Rules Framework and determine which one provides the most flexibility while retaining performance". In an effort to satisfy the first requirement, our team researched three different frameworks, OpenRules, EasyRules, and Drools. In the following report we will discuss our findings of each framework, summarize our findings into a conclusion, and then present our decision at the end.

Open Rules:

Cost – OpenRules licensing policy is dual which makes OpenRules software available to both open source and commercial customers:

- GPL Licenses for Open Source Projects - you have the options of the evaluation version and the complete version. The evaluation version will allow you to install all provided decision models locally, create your own decision models, execute, debug, and analyze them in your own environment. The complete version will allow the same but include documentation, sources, more projects, and a few supporting components.
- Non-GPL Licenses for Commercial Projects – ranges from $3,000 to $13,000 annually.

Documentation – OpenRules provides documentation on its product including step by step installation and usage instructions, web service deployment, database integration, and more.

Support – Technical support is the most important priority of OpenRules, Inc. OpenRules accompanies all downloads of its open source software with email-based technical support. You can pay more to receive faster support or for support during extended business hours. You can also apply for a more discounted support option.

Ease of Use – Following the instructions, using the program is fairly straightforward. Data, datatypes, and rules are inputted into separate excel spreadsheets which the rules engine will use. As long as you follow the format correctly, it should work properly.

Reliability – It has been tested in real world business environments with positive results.

Ease of Program Modification – The evaluation copy of the program isn't modifiable.

External Rules – OpenRules allows for the incorporation of external rules. The primary source of rules for OpenRules is from excel spreadsheets, however, you can bring in rules from Java, a Database, XML, Excel, or a GUI. There is documentation to do such.

Conclusion:

OpenRules has a large base of support and good documentation. It is open source but you must pay a price depending on the licensing you wish to have and the version of the product. From a non-programmer point of view, it is fairly simple to understand how to take their format and incorporate your own rules into it. However, modifying the Java files it is built on is not possible for the evaluation copy. We are unsure as to whether that is possible in the complete version.

EasyRules

Cost - Absolutely no cost is associated with utilizing and incorporating Easy Rules into any Project.

Documentation - Extensive documentation is provided on easyrules.org, and this documentation is maintained by the organization, such that it is always up-to-date and readily available by the developers. The Documentation also features numerous example projects to learn and fully understand the different utilizations of various feature sets of the engine.

Support - There is no known technical support for EasyRules.

Lightweight - The rules engine is incredibly lightweight, in fact it requires zero JAR dependencies which means easy installation. The engine can also function in a standalone or embedded mode.

Infinite Complexity - There is no limit to how complex rules can be, the internet abounds with numerous examples of people utilizing the engine for various complex tasks.

- Rules can be changed, or updated, or even deactivated Dynamically (or programmatically) at run time.
- Easy to program and easy to learn.
- POJO based -(Plain Old Java Object) so it is inherently familiar to anyone who has utilized Java in the past.
- Allows composite (multi part) rules to be created.
- Allows rules to be checked at any time, without firing the rules.
- Allows code to be implemented before the rules are evaluated.

Reasons against Easy Rules:

- Does not support rule chaining.
- Does not support breadth first or depth first rule search.
- Executes all rules “simultaneously”, albeit exactly one time - there is no way to run specific rules in sequence, or to check various rules at alternating frequencies.

- Each rule must be its own class.
- Rules can not be created (or destroyed) dynamically or programmatically.
- Overly Simplified- not many controls over how the engine executes rules.

Conclusion:

In conclusion easy rules would be an acceptable choice for a simple project, it is easy to learn, easy to implement, and is inherently familiar due to its basic Java object structure. The fact that the engine is oversimplified, is actually the primary negative as to why we cannot recommend it for use in our project. The rules engine features critical failures that would be necessary for our use, to develop a fully functional program as the specifications require. The fact that each rule must be it's own class is a major detriment for our project, this is in addition to the fact that rules cannot be created dynamically or programmatically. The fact that the rules cannot be created programmatically means each rule must be a code change, this would prove incredibly impractical for our end-users - yes a workaround could likely be implemented, however this would be a substantial amount of work.

Another key failure of Easy Rules is the fact that it does not permit specific rules to be executed more than once, rather all rules must be executed the same number of times and all at once. Additionally, due to the execution method, the engine does not support rule chaining which was specifically stated as a necessary feature in the requirements document.

Drools

Drools is a business rules management system, with one of its key functionalities being forwards and backwards chaining. Being an open source engine, there is no cost in implementing Drools into a software program.

Developed by jBoss (a subsidiary of Red Hat), Drools has extensive documentation available online on their main website www.drools.org. There are multiple ways one can go about utilizing Drools for their projects:

- Drools Workbench is a web-based UI for creating and managing rules
- Drools Expert is the rules engine which Drools runs off of
- Drools Fusion is the event processing module for the engine
- jBPM is the workflow engine that underlies how the engine processes rules

Cost - Absolutely no cost is associated with utilizing and incorporating Drools into any Project.

Documentation - An extensive documentation resource is made available by the jBoss community. Alongside the documentation, there is a well edited API for not only Drools, but also all of its alternative services such as Workbench and Drools Fusion.

Support - Being fully open sourced, there is a big community following Drools development. This makes troubleshooting issues with implementing its rules engine a lot simpler as there are plentiful resources available for reference. Drools is developed by jBoss who is in turn managed by Red Hat Inc. This alone carries much weight in advocating the benefits of utilizing Drools for the program.

- Declarative implementation - The logic is defined in a declarative way making much easier than writing in application code.
- Data and logic separation
- Centralization of business logic
- Can use different forms of input data for rules.
- Tool Integration - ie. Eclipse Plugin
- Supports rule chaining
- Implements the Rete Algorithm - Faster Speed and Scalability

Conclusion
Drools comes with the support of an always updating extensive documentation provided by the developers jBoss. This is one of the leading reasons to use Drools, as the community behind the rules engine and the documentation can help with any future problems. It is also fully open source and costing nothing to implement making Drools a suitable Rule's Engine. With the ability to separate data and logic, Drools mainly focuses on the business logic allowing for other components to work alongside it. One of our main focuses is rule chaining, or allowing rules to call other rules, which is essential for the task at hand. Drools implements the Rete Algorithm which allows for any frequent change requests, you can add new rules without having to modify the existing rules.

Decision

Based on the information above our group has decided to use Drools as our open source Rules Framework. EasyRules was eliminated early on when we found that it did not provide rule chaining and that every rule had to be its own Java class. If thousands of rules needed to be made for a Command and Control situation on a Navy ship, maintaining all these classes would be very taxing on the memory. That left our decision to either OpenRules and Drools.OpenRules and Drools both have strong technical support, documentation, support rule chaining, and have been tested in real world environments. There is not much difference between the two frameworks in general except for the fact that OpenRules requires a license which requires an annual payment. If our group is able to get full functionality out of Drools then there is no reason to pay for OpenRules. Therefore we have decided to go with Drools in order to create our Rules Based Decision Framework Aid.

Sources

http://openrules.com/licensing.htm
http://openrules.com/documentation.htm
http://openrules.com/support.htm
http://openrules.com/pdf/OpenRulesUserManual.ExternalRules.pdf
http://openrules.com/docs/man_data.html
https://dzone.com/articles/really-simple-powerful-rule
http://www.easyrules.org/index.html
http://www.javaworld.com/article/2071870/enterprise-java/add-a-simple-rule-engine-to-your-spring-based-applications.html
www.drools.org
https://docs.jboss.org/jbpm/v6.3/javadocs/
http://docs.drools.org/6.4.0.Final/drools-docs/html_single/index.html
http://techalpine.com/what-is-a-rule-engine
https://www.tutorialspoint.com/drools/drools_quick_guide.htm