

# Position Paper on Rule Languages for Interoperability

## Authors:

Ravi Sundaar, Aretus Inc. ([rsundaar@aretus.biz](mailto:rsundaar@aretus.biz))

Minerva Hobbs, Fannie Mae ([minerva\\_hobbs@fanniema.com](mailto:minerva_hobbs@fanniema.com))

## Position

Over the last few years, the researchers and practitioners have realized the clear need for rule bases to be distributed and interoperable over the web to realize their true potential. It is our position that, in order for this to happen there needs to be efforts in several different fronts simultaneously.

This document lists some high level requirements in several of the areas that we believe require further efforts.

## Requirements

### ***Representation Language***

A language to represent a rule base must have the following basic requirements.

- We must have a rule representation language that is vendor neutral.
- It must be capable of representing domain objects, their attributes, relationships between objects and rules.
- The representation language must provide for enhancements by various vendors and practitioners. This approach will promote accelerated and decentralized innovation.
- We must be able to export and import a rule base in a vendor neutral language. This is necessary especially when a vendor has its own proprietary language to represent the rule base.
- The rule language must allow for a meta language that describes various aspects of a rule base itself.
- We should be able to group rules using the meta language.
- We must be able to make various groups of rules active or passive conditionally using the meta language.
- We must be able to specify the procedural “flow” (process rules) through a rule base using meta language. This is a practical necessity for most rule bases.

### ***Rule Base Query Language***

Any integration and interaction with a new system starts with a human studying and analyzing the system. For rule bases to be accessible over the Internet, software engineers must have the tools and the means to analyze them.

We must have a query language that would allow an engineer to interrogate and understand various aspects of a rule base. This is akin to someone studying documents, interfaces and using SQL queries to learn how to integrate a database.

- We must have a vendor neutral rule base query language.
- We must be able to query for various aspects of the knowledge base, such as existence of a domain entity, relationship between domain entities, rules that apply to a domain entity and so on.
- The query language should queries at various levels: rule base level, rule group level and individual rule level.
- Query language should be web based (possible Web Services based).

### ***Rule Base Interchange Language***

In order for a rule base to be used by an external application, we need a Rule Base Interface language. The interchange language would support various interactions a rule base and an external application.

- An application must be able to start, stop a rule base and obtain results.
- The interactions should be both synchronous and asynchronous.
- The interactions should be web based (possibly Web Services based).
- The interchange language must support interactions at the rule group level; it must be possible for an application to invoke only the relevant rule groups.
- An application must be able to dynamically modify a rule base for a particular run. For example, assume a company hosting a financial allocation planner with rules. Various organizations should be able to invoke the same rule base with modifications based on their respective policies.
- Rule bases should be able to interchange meta rules for specialization, generalization, overriding, substitutions and so on.

### ***Tools***

One of the key factors for adoption of the standards in rule base language is the availability of tools. Even though the tools themselves do not add to the standards development, they are important for their adoption.

- We may need IDEs (Eclipse, for instance) to have features that will accelerate development of interactions with rule bases. For instance, a software engineer must be able to browse the various ways to interact with a rule base.
- We may need tools that validate the syntax for interactions with a rule base (based on the rule base interchange language grammar).

### **Conclusion**

The requirements listed in this document address various areas of Rules interoperability. As we stated earlier in the document, a multi-pronged approach will help in adoption of the standards that this workshop participants work towards. It is also clear from the document that any work in Rules interoperability requires the participation of people with varied experiences in this field.