Organizing a set of rules

> *This column is the next in a series that provides the reader with best practices on using or choosing a rules engine. The target audience for this series is typically the user of a rule engine, i.e., a programmer or someone with programming skills. All coding examples should be read as pseudo-codeand should be easily translated to a specific target syntax for a rule engine that supports backward and forward chaining in an object-oriented environment.*

We will discuss recommendations on how to organize rules in rule sets. In this description the following concepts are important:

- Rule set: a collection of rules (or rule sets) that are grouped

  The grouping can be done in various ways.
- Infer block: a construct that uses rules to solve a problem

  The solution algorithm may be goal driven (backward chaining) or data driven (forward chaining).
- Inference Task: a set of methods that are used to perform a task, using one or more infer blocks

  Besides the execution of the infer block it also takes care of the preparation and the processing of the results of the infer block.
- Rule service: the collection of inferences, rule sets and rules that make up a service

Infer blocks should be associated with a solution (or sub-solution) of your application. But, although there may be a many-to-many relationship between rule sets and infer blocks, this does not mean that rule sets have to be task- or solution-oriented!

As a matter of fact, it is essential that there be a clear distinction between POSTING rules and EVALUATING rules. This gives us the opportunity to organize rules in a task- or solution-***independent*** way.

I know that, in most projects, the rules are derived in a task-oriented way, so it would be quite awkward to think of another way to group rules than in the way you are doing now in your functional design. Sometimes you can ask your experts to come up with some kind of grouping mechanism. You should always use the grouping mechanism that ‘comes natural’ with your

own work.

My recommendation is to organize rules in rule sets (rule groups) in a way that they can be easily mapped to your functional documentation. For example, if your documentation has paragraphs or sections with ‘related’ rules, you can create rule sets in the same way.

In general you can group rules according to how they are delivered (legislation, contracts, departments, etc.) or how the rules are used (applications, tasks, processes, departments, etc.).

By the way, rule sets may refer to other (sub) rule sets if you wish. It gives you a little more organizational flexibility.

This article was originally published by BRCommunity (link).