

# Drools rules engine

I've been using Drools as a central part of the Identity Management system at Cardiff for about 4 years. And was asked on the last design call (14 April 2010) to write up a summary of this. It does all the following for us:

1. Maps names from different namespaces in different connected applications
2. Makes decisions on what should happen based on the data - e.g. decides what entitlements an account should be given, or what grouper groups it should automatically be a member of

It does increase complexity at the outset, but with the advantage of reducing it long term. The big decision is whether the added complexity is worth it - I'd say that a good guide would be to say that it is if the use of rules is likely to expand.

Major attractions of Drools when we considered it:

1. Fast - far faster than other methods (such as XSLT), this is especially useful as we have thousands of rules
2. Embeddable in Java apps
3. Flexible - we found it more flexible than using lookup tables for namespace mapping
4. Separates the business logic from the application code, meaning that you don't need to be a developer to maintain rules
5. Rules can be written in spreadsheets (as well as many other methods)
6. Rules can be very complicated, but it's always fairly easy to see why something has happened as long as all rules are held within Drools
7. Rules operate on normal JavaBeans, so there's no conversion to/from a specific format
8. Drools rulesets can call functions that are java code in order to manipulate data

What else we've found since using it:

1. Centralised rule repository and versioning available with Drools Guv'nor, which used an SQL database for storage
2. Workflow functionality now available (Drools Flow), which we plan to use to manage the order in which rulesets are applied, but can do much more
3. Temporal reasoning now available (Drools Fusion), allowing the timing of events or groups of events to be used in the application of rules. We're just beginning to explore this and come up with use-cases for which it may be useful. At the moment it seems that monitoring and alerting could be a big area.

Links:

<http://magazine.redhat.com/2008/07/10/introduction-to-drools-rules-fall-from-your-eyes/>  
<http://www.jboss.org/drools>