

Jadex Rules

Jadex Rules is a small lightweight rule engine, which currently employs the well-known Rete algorithm for highly efficient rule matching. Jadex rules is therefore similar to other rule engines like JESS and Drools. Despite the similarities there are also important differences between these systems:

- **Jadex Rules is very small and intended to be used as component of other software.** Even though rules can be specified in a Java dialect as well as (a small variation of) the CLIPS language its primary usage is on the API level. Jadex Rules is currently the core component of the Jadex BDI reasoning engine.
- **Jadex Rules cleanly separates between state and rule representation.** This allows the state implementation as well as the matcher to be flexibly exchanged. Some experiments have e.g. been conducted with a Jena representation. Regarding the matcher, it is planned to support also the Treat algorithm, which has a lower memory footprint than Rete.
- **Jadex Rules pays close attention to rule debugging.** The state as well as the rete engine can be observed at runtime. The rule debugger provides functionalities to execute a rule program stepwise and also use rule breakpoints to stop the execution at those points.

In order to understand how the system can be used you may have a look at the examples code. The examples can be started by simply executing their main method.

Related Projects

Jadex Rules is used within the [Jadex BDI Agent System](#) , which builds a BDI (Belief Desire Intention) agent architecture on top of pure rule kernel.