

# Rule Engines and Agent-Based Systems

**Agostino Poggi**

*Università di Parma, Italy*

**Michele Tomaiuolo**

*Università di Parma, Italy*

## INTRODUCTION

*Expert systems* are successfully applied to a number of domains. Often built on generic *rule-based systems*, they can also exploit optimized algorithms.

On the other side, being based on loosely coupled components and peer to peer infrastructures for asynchronous messaging, *multi-agent systems* allow *code mobility*, adaptability, easy of deployment and reconfiguration, thus fitting distributed and dynamic environments. Also, they have good support for domain specific *ontologies*, an important feature when modeling human experts' knowledge.

The possibility of obtaining the best features of both technologies is concretely demonstrated by the integration of *JBoss Rules*, a rule engine efficiently implementing the *Rete-OO algorithm*, into *JADE*, a *FIPA-compliant multi-agent system*.

## BACKGROUND

### Rule Engines

The advantages of *rule-based systems* over procedural programming environments are well recognized and widely exploited, above all in the context of business applications. Working with rules helps keeping the logic separated from the application code: it can be modified by non-developers and, being centralized in one point, it can be analyzed and validated. *Rule engines* are often well optimized, being able to efficiently reduce the number of rules to match against the updated knowledge base.

*Rule-based systems* can also be augmented with ideas and techniques developed in other research fields, leading for example to fuzzy rule-based systems, which exploit fuzzy logic to deal with imprecision and uncertainty about the knowledge base. Moreover,

sometimes these systems are coupled with genetic algorithms and evolutionary programming to generate complex classifiers.

One of the most notable application of rule-based systems are *expert systems*, where the rule set is a representation of an expert's knowledge. In such systems, the AI (Artificial Intelligence) is supposed to perform in a similar manner to the expert, when exposed to the same data.

Among the different mechanisms to implement a rule-engine, *Rete algorithm* (Forgy, 1982) has gained more and more popularity, mainly thanks to the high degree of optimization that can be obtained. At NASA Johnson Space Center, *Rete algorithm* was implemented in a whole generation of rule engines. OPS5 was soon replaced by its descendant, ART, and in 1984 by the more famous CLIPS.

Nowadays, one of the most widespread engines implementing *Rete* is *Jess* (Friedman-Hill, 2000), at first developed as a Java port of CLIPS at Sandia National Laboratories in late 1990s. *Jess* has also been widely adopted by the agent community to realize rule-based agent systems (Cardoso, 2007).

A different yet promising rule-engine is *JBoss Rules* (Proctor, Neale, Frandsen, Griffith & Tirelli, 2007), formerly *Drools*. It is a quite new, but already well known, freeware tool implementing so-called *Rete-OO algorithm*.

Its open-source availability is a clear advantage over *Jess*, but an even greater advantage is due to the implementation of a particular adaptation of the *Rete algorithm* for the object-oriented world, rather than a literal one. This way, the burden of integrating the rule-engine and application rules with existing external objects is greatly reduced. In fact, *JBoss Rules* uses plain Java objects to represent rules and facts, which can be modified through their public methods and properties. Rules can be specified through an appropriate syntax, or through xml structures, and their conditions and consequences can be expressed using different scripting

Figure 1. Basic example from the JBoss Rules handbook

```

package org.drools.examples

import org.drools.examples.HelloWorldExample.Message;

rule "Hello World"
when
    m : Message( status == Message.HELLO, message : message )
then
    System.out.println( message );
    m.setMessage( "Goodbye cruel world" );
    m.setStatus( Message.GOODBYE );
    update( m );
end

rule "GoodBye"
no-loop true
when
    m : Message( status == Message.GOODBYE, message : message )
then
    System.out.println( message );
end

```

languages, as Python, Groovy and Java. Instead *Jess* only accepts rules written in the CLIPS language, thus requiring developers to learn a new Lisp-like language and deploy additional efforts to adapt it to their object-oriented development environment.

## Agent-Based Systems

*Multi-agent systems* (MAS) show some complementary features which can be useful in many rule-based application, above all asynchronous interaction protocols and semantic languages. In multi-agent systems, in fact, many intelligent agents interact with each other. The agents are considered to be autonomous entities, and their interactions can be either cooperative or selfish (i.e. they can share a common goal, as in a production line, or they can pursue their own interests, as in an open marketplace).

The *Foundation for Intelligent Physical Agents* (FIPA, 2002) develops open specifications, to support interoperability among agents and agent-based applications. Specifications for infrastructures include a communication language for agents, services for agents, and they anticipate the management of domain-specific

*ontologies*. A set of application domains is also specified, including personal assistance for travels, network management, electronic commerce, distribution of audio-visual media. At the core of FIPA model there's the communication among agents; in particular it describes how the agents can exchange semantically-meaningful messages with the aim of completing activities required by the overall application.

Various implementations of FIPA-compliant platforms exist (FIPA implementations, 2003). Among them, *JADE* (Bellifemine, Caire, Poggi & Rimassa, 2003) has gained popularity during the years, while more and more core functionalities and third-party plug-ins were being developed. Currently it supports most of the infrastructure related *FIPA* specifications, like transport protocols, message encoding, and white and yellow pages agents. Moreover, it has various tools that ease agent debugging and management.

The possibility of using rules to realize agent systems seems to be promising. On the one hand, rules have been shown suitable to define abstract and real agent architectures and have been used for realizing so-called "*rule-based agents*", that is, agents whose behaviour and/or knowledge is expressed by means of

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/rule-engines-agent-based-systems/10423](http://www.igi-global.com/chapter/rule-engines-agent-based-systems/10423)

## Related Content

---

### AI in Health and Diagnostics

Nidhi Ashokbhai Jodhwani and Sanketkumar Jagdishbhai Ahir (2022). *Handbook of Research on Lifestyle Sustainability and Management Solutions Using AI, Big Data Analytics, and Visualization* (pp. 1-17).

[www.irma-international.org/chapter/ai-in-health-and-diagnostics/298365](http://www.irma-international.org/chapter/ai-in-health-and-diagnostics/298365)

### Industry 5.0: A Macroperspective Approach

Saurabh Tiwari, Prakash Chandra Bahuguna and Jason Walker (2022). *Handbook of Research on Innovative Management Using AI in Industry 5.0* (pp. 59-73).

[www.irma-international.org/chapter/industry-50/291461](http://www.irma-international.org/chapter/industry-50/291461)

### An RHMIoT Framework for Cardiovascular Disease Prediction and Severity Level Using Machine Learning and Deep Learning Algorithms

Sibo Prasad Patro and Neelamadhab Padhy (2022). *International Journal of Ambient Computing and Intelligence* (pp. 1-37).

[www.irma-international.org/article/an-rhmiot-framework-for-cardiovascular-disease-prediction-and-severity-level-using-machine-learning-and-deep-learning-algorithms/311062](http://www.irma-international.org/article/an-rhmiot-framework-for-cardiovascular-disease-prediction-and-severity-level-using-machine-learning-and-deep-learning-algorithms/311062)

### Event-Based Social Networking System With Recommendation Engine

G. Manikandan, Reuel Samuel Sam, Steven Frederick Gilbert and Karthik Srikanth (2024). *International Journal of Intelligent Information Technologies* (pp. 1-16).

[www.irma-international.org/article/event-based-social-networking-system-with-recommendation-engine/334232](http://www.irma-international.org/article/event-based-social-networking-system-with-recommendation-engine/334232)

### A New Technology of Fuzzy Logic in Machinery Monitoring

(2018). *Fuzzy Logic Dynamics and Machine Prediction for Failure Analysis* (pp. 1-15).

[www.irma-international.org/chapter/a-new-technology-of-fuzzy-logic-in-machinery-monitoring/197316](http://www.irma-international.org/chapter/a-new-technology-of-fuzzy-logic-in-machinery-monitoring/197316)