Sharing rules between JBoss and Jena

Oana Nicolae¹  Ion Mircea Diaconescu¹  Adrian Giurca¹  Gerd Wagner¹

¹Brandenburgische Technische Universität Cottbus, Germany

Timisoara, Romania
Outline

1. **Introduction**
   - Rule Modeling and Rule Interchange

2. **JBossRules to JenaRules**
   - Mapping Vocabulary
   - Mapping Rules
   - Discussions about the limitations of the proposed interchange

3. **Conclusions**
There are different rule target platforms and languages:

- Object Oriented Rule Languages (JBossRules, JRules (ILog), Fair Isaac, Oracle Business Rules)
- Semantic Web Rule Languages (JenaRules, SWRL)
- Artificial Intelligence Rule Languages (Jess, Prolog)

Rule-based vendor tools are seriously competed by open-source rule engines/platforms.

Interoperability initiatives fight for a commonly agreed Rule Interchange Format.
A Rule Markup Language

- Allows reuse, interchange and publication of rules.
- Provides a platform independent syntax in order to represent rule languages that come from different areas.
- Must cover different rule languages having different intended semantics.
- Needs a vocabulary and expressions to encode rules.

A Tutorial about Rule Interchange

Our work on Rule Interchange

- Our rule interchange work addresses JBossRules\(^1\) 3.0.6 as source platform and JenaRules\(^2\) 2.5.3 as a target platform, bridging this way an Object-Oriented rule language to a Semantic Web rule language.
- We use R2ML 0.4 as Interchange Language Format.

---

R2ML - REWERSE I1 Rule Markup Language
http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=R2ML

RIF - Rule Interchange Format
http://www.w3.org/2005/rules/

---

\(^1\)JBossRules - [http://labs.jboss.com/jbossrules/](http://labs.jboss.com/jbossrules/)
Mapping Vocabulary

- JBossRules are build on top of Java vocabularies which consists of Java beans.
- An R2ML rule may refers to a vocabulary which can be R2ML own vocabulary or an imported one (such as RDF(S)\(^3\) and OWL\(^4\)). All standard Java datatypes maps into XML Schema datatypes (See JSR 31\(^5\)). Any Java qualified class names will be translated in R2ML into appropriate qualified names, as a reference to the class name, together with their corresponding namespace declarations.
- JenaRules Vocabulary is expressed by RDF(S).

---

\(^3\)RDF(S) - http://www.w3.org/TR/rdf-schema/
\(^4\)OWL - http://www.w3.org/2004/OWL/
JBossRules as source language

- Rule engine based on an enhanced version of Rete algorithm.
- Open-source, object-oriented production rules system written entirely in Java language.
- Separation of the business logic (the rules) from business data (the facts).
- The runtime provides dynamic assertion and remove of rules.
- Employs conflict resolution using salience rule attribute and LIFO.
- Light and easy to understand syntax (i.e. DRL syntax); uses Java to express field constraints, functions and consequences.
- Easy to integrate with the mainstream JEE5 technologies.
- Collects complex decision-making logic and works with large data sets.
JenaRules as target language

- Open-source, Java framework for building Semantic Web applications
- Use RDF as data (desirable in Semantic Web since people works with Atom, Microformats, FOAF etc )
- Reasoner supports rule-based inference over RDF graphs and provides forward chaining (RETE), backward chaining (Datalog-based) and a hybrid execution model
Mapping of Conditions

JBossRules Implementation

when
driver:Driver(numberOfAccidents > 2)

R2ML Serialisation

<\texttt{r2ml:conditions}>
  <\texttt{r2ml:DatatypePredicateAtom} \texttt{r2ml:datatypePredicateID}="\texttt{swrlb:greaterThan}"
    \texttt{r2ml:dataArguments}>
    <\texttt{r2ml:AttributeFunctionTerm} \texttt{r2ml:attributeID}="\texttt{userv:Driver.numberOfAccidents}"
      \texttt{r2ml:contextArgument}>
      <\texttt{r2ml:ObjectVariable} \texttt{r2ml:name}="\texttt{driver}" \texttt{r2ml:classID}="\texttt{userv:Driver}" />
    </\texttt{r2ml:contextArgument}>
  </\texttt{r2ml:AttributeFunctionTerm}>
  <\texttt{r2ml:TypedLiteral} \texttt{r2ml:lexicalValue}="2" \texttt{r2ml:datatypeID}="\texttt{xs:integer}" />
</\texttt{r2ml:conditions}>

JenaRules Implementation

(?\texttt{driver \texttt{userv:numberOfAccidents} ?numberOfAccidents})
greaterThan(?\texttt{numberOfAccidents},2)
Sharing rules between JBoss and Jena

**JBossRules Implementation**

```java
$driver.isHighRiskDriver(true);
modify($driver);
```

**R2ML Serialisation**

```xml
<r2ml:producedAction>
  <r2ml:AssignActionExpression
    r2ml:propertyID="isHighRiskDriver">
    <r2ml:contextArgument>
      <r2ml:ObjectVariable r2ml:name="driver"
        r2ml:classID="userv:Driver"/>
    </r2ml:contextArgument>
    <r2ml:TypedLiteral r2ml:lexicalValue="true"
      r2ml:datatypeID="xs:boolean"/>
  </r2ml:AssignActionExpression>
</r2ml:producedAction>
```

**JenaRules Implementation**

```java
(?driver userv:isHighRiskDriver 'true'^^xs:boolean)
```
Neither JBossRules nor JenaRules don’t provide a semantics of their languages. The correctness of the proposed translation was established by testing rules.

The tests are available in the application online version\(^6\).

We translate the JBoss production rules into JenaRules implementation via R2ML and execute those rules based on analogous facts from the Working Memory.

The correctness of the translation implies the same obtained results regarding the facts from Working Memory.

\(^6\text{UServ} - \text{http://oxygen.informatik.tu-cottbus.de/userv/}\)
About Interchange Limitations

- Translation from PSM (JBossRules) to PIM (R2ML) implies the access to the rules vocabulary (Java beans classes).

- JBossRules PR actions find their mapping in the R2ML PR Actions\(^7\). Since JBossRules allows free Java code in the rules action part it cannot be completely mapped to R2ML actions. The new JBoss version (4.0.1) comes with a specific expression language for encoding actions i.e. MVEL\(^8\). A study of mapping from MVEL to R2ML is necessary.

- JenaRules could not represent function calls as arguments for an operation call.

- JenaRules can represent just binary atoms (and built-ins) while R2ML can represent any logical atom.

\(^7\)OMG PRR - [http://www.omg.org/docs/bmi/07-03-05.pdf](http://www.omg.org/docs/bmi/07-03-05.pdf)

\(^8\)MVEL, [http://mvel.codehaus.org/](http://mvel.codehaus.org/)
This paper aims to provide a brief, informal description of the JBossRules to JenaRules translation, based on the R2ML markup language.

It outlines a way to interchange rules between Object-Oriented rule languages and Semantic Web rule languages.

Our work has followed the principles initiated on rules inter-operability such as W3C (i.e. RIF), OMG’s standards and EU network of excellence REWERSE (i.e. R2ML).
Our work includes

- **R2ML** - http://oxygen.informatik.tu-cottbus.de/reverse-il/?q=R2ML
- **Interchange Web Service** - http://oxygen.informatik.tu-cottbus.de/reverse-il/?q=node/27
- **Strelka** - http://oxygen.informatik.tu-cottbus.de/reverse-il/?q=Strelka
- **Use Cases** - http://oxygen.informatik.tu-cottbus.de/reverse-il/?q=node/32