RDF/RuleML Interoperability

W3C Workshop on Rule Languages
for Interoperability
Position Paper, 27-28 April 2005

Authors:

Harold Boley[1], Jing Mei[2],
Michael Sintek[3], Gerd Wagner[4]

[1]NRC, [2]PKU, [3]DFKI, [4]BTU

Contents

1 Introduction

- RuleML has been developed increasingly closer towards RDF, with an eye also on F-logic and Description Logic

- This is already reflected by RuleML/RDF tools and a RuleML/N3 alignment

- The newest RuleML release as described here will further simplify RDF interoperability regarding webizing, typing, and blank nodes

- It also integrates two negations, as required for the Semantic Web

- On this basis, F-logic/TRIPLE rules and SWRL rules are being introduced

2 RDFizing RuleML

- At the RDF Interest Group face to face meeting in Feb 2001 the Rule Markup Initiative was presented with a focus on RDF Relationships and DTD Modularization

- This section shows the RDFizing of RuleML, using the new RuleML XML syntax

2.1 RDF Triples and Rules as RuleML Sublanguages (1)

- Before Object-Oriented (OO) RuleML became available, the family of sublanguages of RuleML had to use **(subject, predicate, object)** triples for representing RDF as ground (variablefree) binary atoms **predicate(subject, object)**, where **predicate** and **subject** needed to be webized and **object** was optionally webized

- RDF rules could then be represented by allowing variables inside binary atoms

2.1 RDF Triples and Rules as RuleML Sublanguages (2)

- The slots of OO RuleML permit to express 'subject-centered' RDF descriptions without duplicating a subject for all of its triples

- These descriptions are *anchored* in the Web through URIs

- To integrate RDF more deeply with other aspects of RuleML, and to reduce the number of sublanguages, the binary atoms are no longer distinguished in the newest RuleML

2.2 RDF-Like Webizing of RuleML (1)

■ RuleML allows an individual constant (Ind) to contain web labels and web references, leading to this RDF correspondence:

■ The ***subject*** of an RDF triple, or the about attribute in an `rdf:Description`, becomes a RuleML ***role tag*** called **oid** containing an **Ind** with a **wlab** (web label) attribute

2.2 RDF-Like Webizing of RuleML (2)

- An RDF *predicate* or *property* becomes the <u>first child</u> of a RuleML **slot**, which is an **Ind** with a **wref** (web reference) attribute

- An RDF *object* becomes the <u>second child</u> of a RuleML **slot**, which for an RDF **literal** is a RuleML **Data** element and for an RDF **resource** again is a **wref**-attributed **Ind**

2.2 RDF-Like Webizing of RuleML (3)

Illustrating the general correspondence (omitting namespace declarations):

```
<rdf:Description about="http://www.w3.org">
  <dc:title>World Wide Web Consortium</dc:title>
  <eg:director rdf:resource="http://www.w3.org/People/Berners-Lee"/>
</rdf:Description>
```

```
<Atom>
  <oid><Ind wlab="http://www.w3.org"/></oid>
  <slot>
    <Ind wref="dc:title"/>
    <Data>World Wide Web Consortium</Data>
  </slot>
  <slot>
    <Ind wref="eg:director"/>
    <Ind wref="http://www.w3.org/People/Berners-Lee"/>
  </slot>
</Atom>
```

2.3 RDF Types and RDF Schema in RuleML

- The **rdf:type** of a resource in RuleML is regarded as the *relation* of its describing atom.
  Again via a web reference, it points to an RDF Schema definition

- Illustrating the **type**-**relation** correspondence (omitting namespace declarations):

```
<rdf:Description about="http://www.w3.org">
 <rdf:type resource="http://schema.org/StandardsBody"/>
 <dc:title>World Wide Web Consortium</dc:title>

 . . .
</rdf:Description>
```

```
<Atom>
 <oid><Ind wlab="http://www.w3.org"/></oid>
 <opr><Rel wref="http://schema.org/StandardsBody"/></opr>
 <slot><Ind wref="dc:title"/><Data>World Wide Web Consortium</Data></slot>

 . . .
</Atom>
```

- For multiple types of a resource, an atom with multiple relations can be used, which provides a useful shortcut even in the absence of an oid

2.4 Blank Nodes as Skolem Constants

- Blank Nodes have recently been introduced to RuleML for complete RDFizing:
  Skolem constants are marked up as

  <Skolem/> (anonymous) or

  <Skolem>...</Skolem> (named),

  and adopt the Yang/Kifer model theory [5]

3 Two Kinds of Negation in RDF/RuleML

While the previous section showed how RuleML caught up with RDF expressiveness, the current section discusses RuleML sublanguages also proposed as RDF extensions

In his visionary article [2], Tim Berners-Lee identified two fundamental issues for the Semantic Web:

- Negation and inconsistency handling
- Open-world versus closed-world assumptions

3 Two Kinds of Negation in RDF/RuleML *cont'd*

RDF does not even have any kind of negation, while OWL has a limited form of negation only

Neither the current RDF nor OWL are able to deal with Closed-World assumptions, nor do they have any feature for handling inconsistency

3 Two Kinds of Negation in RDF/RuleML *cont'd*

In [3], Tim Berners-Lee says: "We remove the centralized concepts of absolute truth, total knowledge, and total provability, and see what we can do with limited knowledge"

This is exactly what has been done in RuleML by providing two kinds of negation on the basis of partial logic. Unlike classical logic, which is the basis of OWL, partial logic allows to handle *truth value gaps* (partiality) and *truth value clashes* (inconsistency)

3 Two Kinds of Negation in RDF/RuleML *cont'd*

- RuleML includes the computational construct of negation-as-failure (**NAF**) for expressing local completeness (Closed-World) assumptions

- And it includes a more conservative negation connective (**NEG**), also called *strong negation*, which corresponds to Kleene negation in 3-valued logic or to Boolean negation, depending on the semantic nature of the predicate being negated

- With these computational logic facilities, RuleML supports more real-world knowledge representation tasks than RDF and OWL, which are based on the highly idealized classical (2-valued) logic
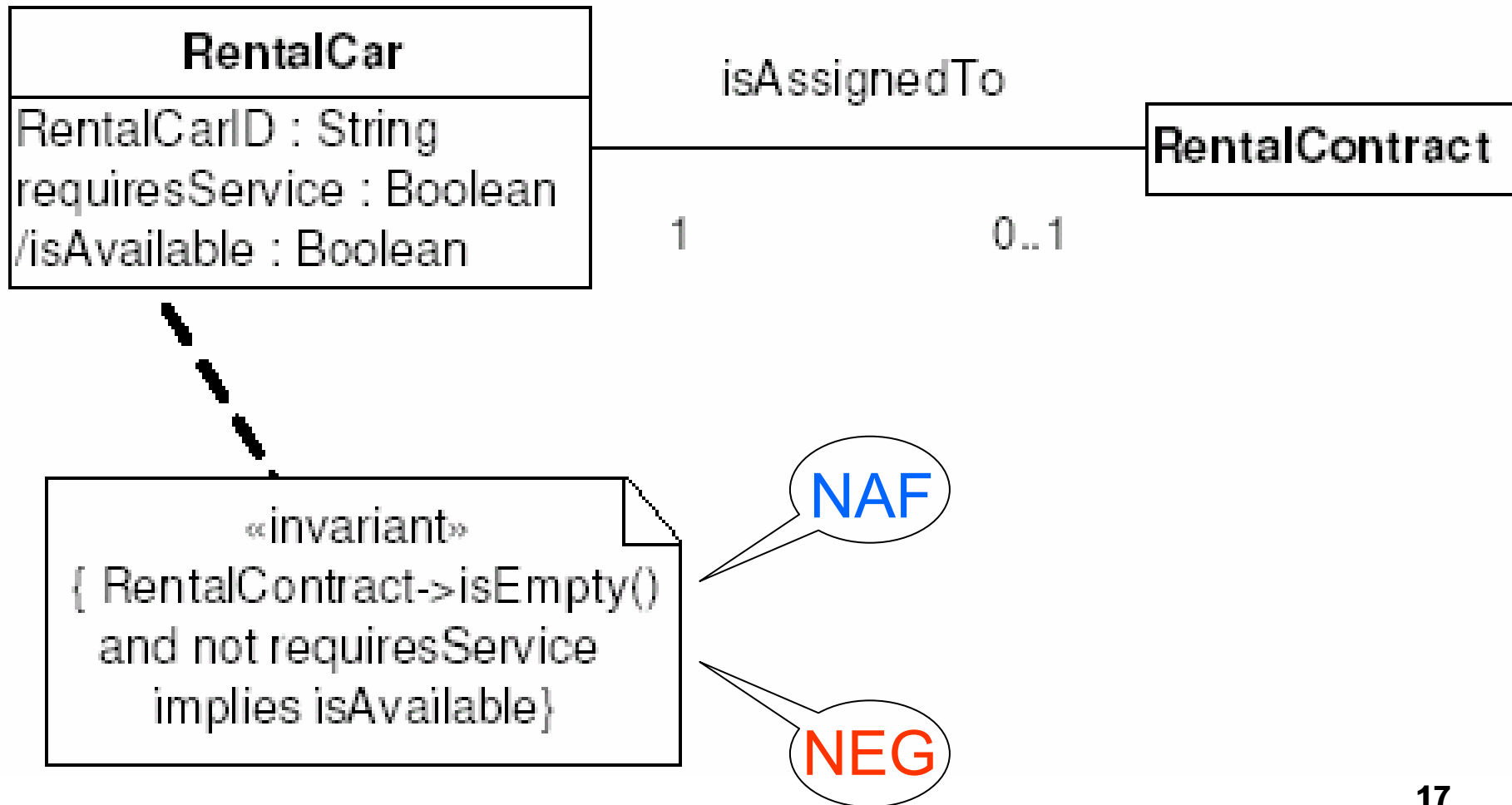
3 Two Kinds of Negation in RDF/RuleML *cont'd*

- These semantics are reflected by RuleML sublanguage schemas with NAF, NEG, and both. RDF can be extended by adding both forms of negation, too, as has been shown in [1]

- The following is an example of a rule using both kinds of negation

- Here, the predicate "requiresService" does not allow Closed-World inference, while the predicate "is AssignedToRentalContract" does allow it, and, hence, the former is negated with NEG and the latter with NAF

3 Two Kinds of Negation in RDF/RuleML *cont'd*
A Business Rule Example in OCL

An OCL invariant defining the derived attribute *isAvailable* of the class RentalCar

RentalCar
RentalCarID : String requiresService : Boolean /isAvailable : Boolean

isAssignedTo

RentalContract

1   0..1

«invariant»
{ RentalContract->isEmpty()
and not requiresService
implies isAvailable}

NAF

NEG

A Business Rule Example in RuleML

```
<Implies>
 <body>
  <And>
   <Atom>
    <Rel>RentalCar</Rel>
    <Var>Car</Var>
   </Atom>
   <Neg>
    <Atom>
     <Rel>requiresService</Rel>
     <Var>Car</Var>
    </Atom>
   </Neg>
   <Naf>
    <Atom>
     <Rel>isAssignedToRentalContract</Rel>
     <Var>Car</Var>
    </Atom>
   </Naf>
  </And>
 </body>
```

```
  <head>
   <Atom>
    <Rel>isAvailable</Rel>
    <Var>Car</Var>
   </Atom>
  </head>
 </Implies>
```

18

4 F-logic And TRIPLE Rules in RDF/RuleML

- TRIPLE is seen as part of the RuleML Initiative

- But: we believe a *human-readable* syntax is very important for a rule language, so TRIPLE uses a variant of F-logic syntax, and not XML

- To allow TRIPLE rules to be exchanged between agents (some of which don't speak TRIPLE), RuleML comes into play

4 F-logic And TRIPLE Rules in RDF/RuleML *cont'd*

- **Two options here:**
  - ☐ Map TRIPLE to Horn logic (which is done anyway for enactment) and represent these in RuleML
  - ☐ Extend RuleML to support TRIPLE (or at least crucial parts of it) natively
- **First option does not allow true round-trip**
- **Therefore, RuleML has been or will be extended to support:**
  - ☐ RDF statements (OO RuleML)
  - ☐ First-Order Logic (FOL RuleML)
  - ☐ Contexts/Modules (*planned*)

4 F-logic And TRIPLE Rules in RDF/RuleML *cont'd*

- With the advent of OO RuleML [4], *molecules*, the most fundamental feature of F-logic (and its early RDF-compliant version, SiLRI) and TRIPLE, became representable

- For TRIPLE's context (or model/view) feature, which allows RDF statements to be grouped and "@"-associated with a name (similar to contexts in Cwm or named graphs), a RuleML **module** (or isTrueIn) **role** (on Atom, Implies, etc.) is proposed in analogy to the oid role, which can contain an Individual

TRIPLE Example:

- The following example shows the proposed RuleML serialization for a TRIPLE molecule about W3C in a module context (omitting namespace declarations):

- 'http://www.w3.org':"[rdf:type -> s:StandardsBody;
  dc:title -> "World Wide Web Consortium";
  eg:director -> p:'Berners-Lee']**@c:mod1**.

- ```
  <Atom>
   <module><Ind wref="c:mod1"/></module>
   <oid><Ind wlab="http://www.w3.org"/></oid>
   <opr><Rel wref="s:StandardsBody"/></opr>
   <slot>
     <Ind wref="dc:title"/>
     <Data>World Wide Web Consortium</Data>
   </slot>
   <slot>
     <Ind wref="eg:director"/>
     <Ind wref="p:Berners-Lee"/>
   </slot>
  </Atom>
  ```

5 SWRL Rules in RDF/RuleML

- In the [SWRL submission to W3C](#) an RDF Schema for SWRL ([swrl.rdf](#)) was published to provide a partial description of the syntax of SWRL. A corresponding RuleML version has recently been provided as [swrl.ruleml](#) (with a KQML-like top-level performative 'Assert')

- SWRL rules in OWL RDF/XML syntax are interoperable with an RDFized RuleML version, where, e.g., the IndividualPropertyAtom element corresponds to a normal Atom element with a swrl:propertyPredicate slot

5 SWRL Rules in RDF/RuleML *cont'd*

- Illustrating SWRL Example 6.1-1 in RDFized RuleML (omitting namespace declarations):

```
<Implies>
 <oid><Ind wlab="http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/#Example6.1-1"/></oid>
 <body>
  <And>
   <Atom>
    <opr><Rel wref="swrl:IndividualPropertyAtom"/></opr>
    <slot><Ind wref="swrl:propertyPredicate"/><Ind wref="eg:hasParent"/></slot>
    <slot><Ind wref="swrl:argument1"/><Var wref="eg:x1"/></slot>
    <slot><Ind wref="swrl:argument2"/><Var wref="eg:x2"/></slot>
   </Atom>
   <Atom>
    <opr><Rel wref="swrl:IndividualPropertyAtom"/></opr>
    <slot><Ind wref="swrl:propertyPredicate"/><Ind wref="eg:hasSibling"/></slot>
    <slot><Ind wref="swrl:argument1"/><Var wref="eg:x2"/></slot>
    <slot><Ind wref="swrl:argument2"/><Var wref="eg:x3"/></slot>
   </Atom>
   <Atom>
    <opr><Rel wref="swrl:IndividualPropertyAtom"/></opr>
    <slot><Ind wref="swrl:propertyPredicate"/><Ind wref="eg:hasSex"/></slot>
    <slot><Ind wref="swrl:argument1"/><Var wref="eg:x3"/></slot>
    <slot><Ind wref="swrl:argument2"/><Ind wref="eg:male"/></slot>
   </Atom>
  </And>
 </body>
 <head>
  <Atom>
   <opr><Rel wref="swrl:IndividualPropertyAtom"/></opr>
   <slot><Ind wref="swrl:propertyPredicate"/><Ind wref="eg:hasUncle"/></slot>
   <slot><Ind wref="swrl:argument1"/><Var wref="eg:x1"/></slot>
   <slot><Ind wref="swrl:argument2"/><Var wref="eg:x3"/></slot>
  </Atom>
 </head>
</Implies>
```

6 Conclusions

- **This paper demonstrates**
  - ☐ The high level of RDF/RuleML interoperability achieved
  - ☐ Extensions in close reach from RDF and RuleML

7 References

- 1. A. Analyti, G. Antoniou, C. V. DamÃ¡sio and G. Wagner. Negation and Negative Information in the W3C Resource Description Framework. *Annals of Mathematics, Computing & Teleinformatics*, vol 1, no 2, 2004, pp 25-34.

- 2. T. Berners-Lee. Design issues - architectual and philosophical points. Personal notes, 1998. Available at http://www.w3.org/DesignIssues/.

- 3. T. Berners-Lee. What the semantic web can represent. Personal notes, 1998. Available at http://www.w3.org/DesignIssues/RDFnot.html.

- 4. H. Boley, B. Grosof, M. Kifer, M. Sintek, S. Tabet, G. Wagner. Object-Oriented RuleML, 2004. Available at http://www.ruleml.org/indoo.

- 5. G. Yang and M. Kifer. Reasoning about Anonymous Resources and Meta Statements on the Semantic Web. Journal of Data Semantics, 2004.