Best viewed with PPT 2000

RuleML Meets RDF: Triples, Rules, and Taxonomies

www.daml.org

www.ruleml.org

www.w3.org/RDF

Harold Boley*, NRC IIT e-Business
Benjamin Grosof, MIT Sloan
(with help from Bruce Spencer,
Steve Ross-Talbot, Said Tabet,
and Gerd Wagner)

* On leave from DFKI GmbH

Joint Committee Meeting
5 November 2002

Introduction

- Increased mutual RuleML-RDF(S) interest:
  - RDF and RDF Schema need **rules** for metadata and taxonomy deduction, transformation, etc.; so rules should be interchangeable much like RDF(S) itself
  - RuleML rules need **types** for constraining variables, which should be able to reuse the growing taxonomic vocabularies in the Semantic Web

- In the following we treat three RuleML-RDF(S) topics:
  - RDF triples and rules in RuleML
  - RDF formats for RuleML rules
  - RDFS taxonomies for typed RuleML

Overview of RDF Triples & Rules in RuleML

- RuleML 0.8 uses
  - *RDF triples* as special binary facts and
  - *RDF rules* over such facts

- Both are defined as part of the hierarchy of RuleML DTDs

RuleML 0.8: RDF Triples as Binary Facts

- *RDF triples* become special binary facts where the **relation** and **first argument** must be urirefs, and the **second argument** can be urirefs or literals

"**http://www.w3.org/Home/Lassila**   **has creator**   **Ora Lassila**"

```
<fact>
 <_head>
  <atom>
    <_opr>
     <rel href="http://dublincore.org/documents/dces/index.shtml.rdf#Creator"/>
    </_opr>
    <ind href="http://www.w3.org/Home/Lassila"/>
    <ind>Ora Lassila</ind>
  </atom>
 </_head>
</fact>
```

Original online

RuleML 0.8: RDF Rules Over Triple Facts

- *RDF rules* over triple facts can prove implicit triples, top-down, or can derive new triples, bottom-up

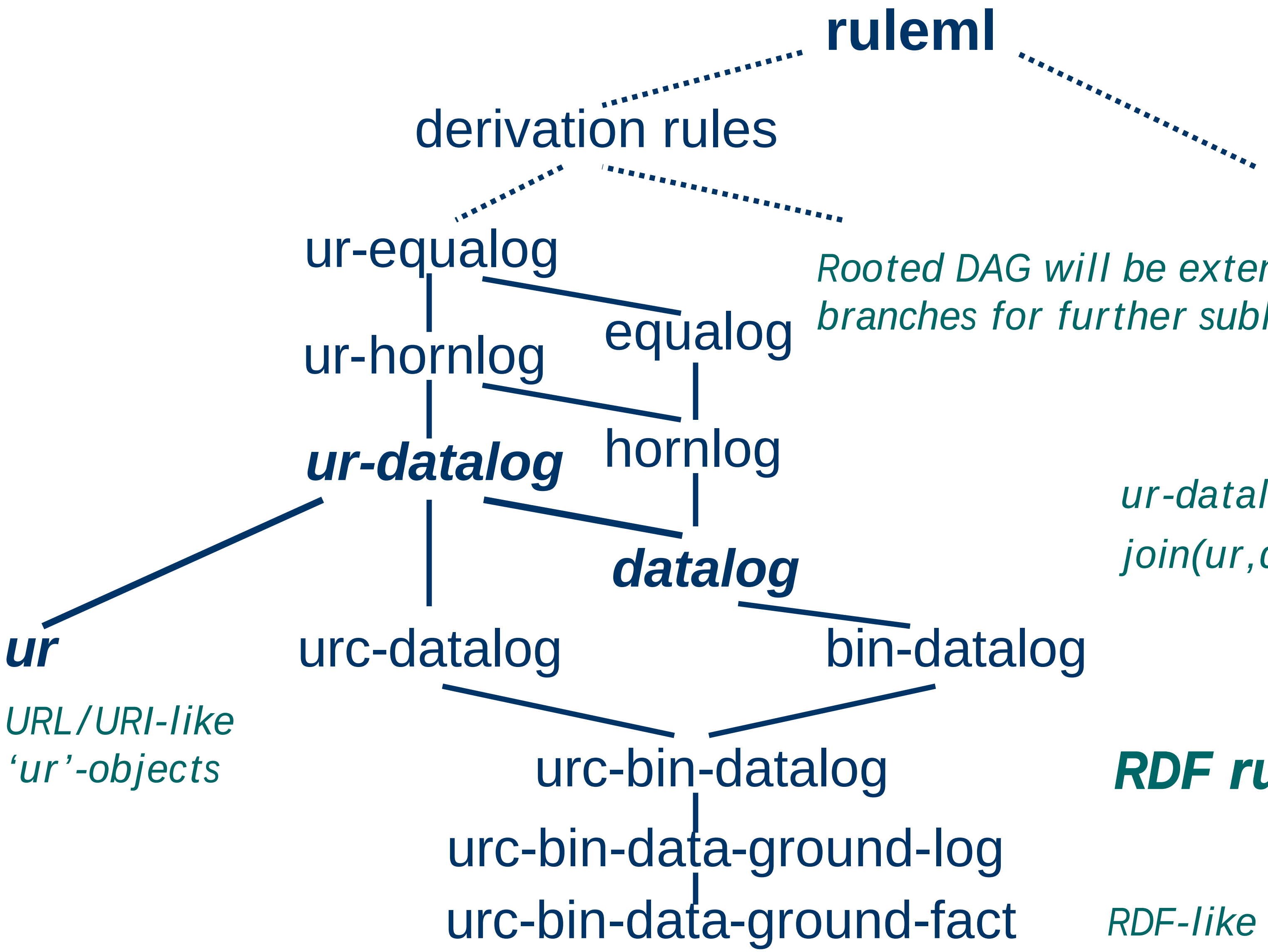
IF "**Page** has creator **Person**" THEN "**Page** was accessed by **Person**"

```
<imp>
 <_body>
  <atom>
    <_opr>
      <rel href="http://dublincore.org/documents/dces/index.shtml.rdf#Creator"/>
    </_opr>
    <var>Page</var>
    <var>Person</var>
  </atom>
 </_body>
 <_head>
  <atom>
    <_opr>
      <rel href="http://logging.org/vocabulary/xyz.rdf#Accessed"/>
    </_opr>
    <var>Page</var>
    <var>Person</var>
  </atom>
 </_head>
</imp>
```

Structure of the RuleML DTD Hierarchy

- Our system of DTDs (current version: 0.8) uses a modularization approach similar to XHTML in order to **accomodate** the various **rule subcommunities**
- The evolving hierarchy of RuleML DTDs forms a *partial order* with **ruleml** as the greatest element (a **ruleml**-rooted DAG) -- many ‘smallest’ elements
- Each DTD node in the hierarchy (conformance “lattice”) corresponds to a specific RuleML sublanguage, **syntactically and semantically**:
  - ‘Union’ (*join*) of sublanguages reached via outgoing links: to smaller or equal nodes below
  - ‘Intersection’ (*meet*) of sublanguages via incoming links: from greater or equal nodes above

The Module Hierarchy of RuleML DTDs



ruleml

derivation rules

ur-equalog

ur-hornlog

equalog

ur-datalog

hornlog

datalog

ur

urc-datalog

bin-datalog

urc-bin-datalog

urc-bin-data-ground-log

urc-bin-data-ground-fact

Rooted DAG will be extended with branches for further sublanguages

ur-datalog = join(ur,datalog)

URL/URI-like ‘ur’-objects

RDF rules

RDF-like triples

Overview of RDF Formats for RuleML Rules

- An experimental translator for the XML-based RuleML 0.7 to RDF has been available in XSLT:
  This was the first RuleML in RDF
- The current RuleML 0.8 stands in a direct RDF Context:
  It integrates the XML and RDF data models
- Michael Sintek has implemented translators between Prolog and an RDF-based RuleML 0.8
- Massimo Paolucci used this RDF RuleML in DAML-S Semantic Matchmaking for Web Services Discovery to describe constraints related to input and output, and also preconditions and effects for planning
- We recently further developed RDF RuleML 0.8 using the W3C RDF Validation Service:
  http://www.w3.org/RDF/Validator/

From Natural Language to Horn Logic

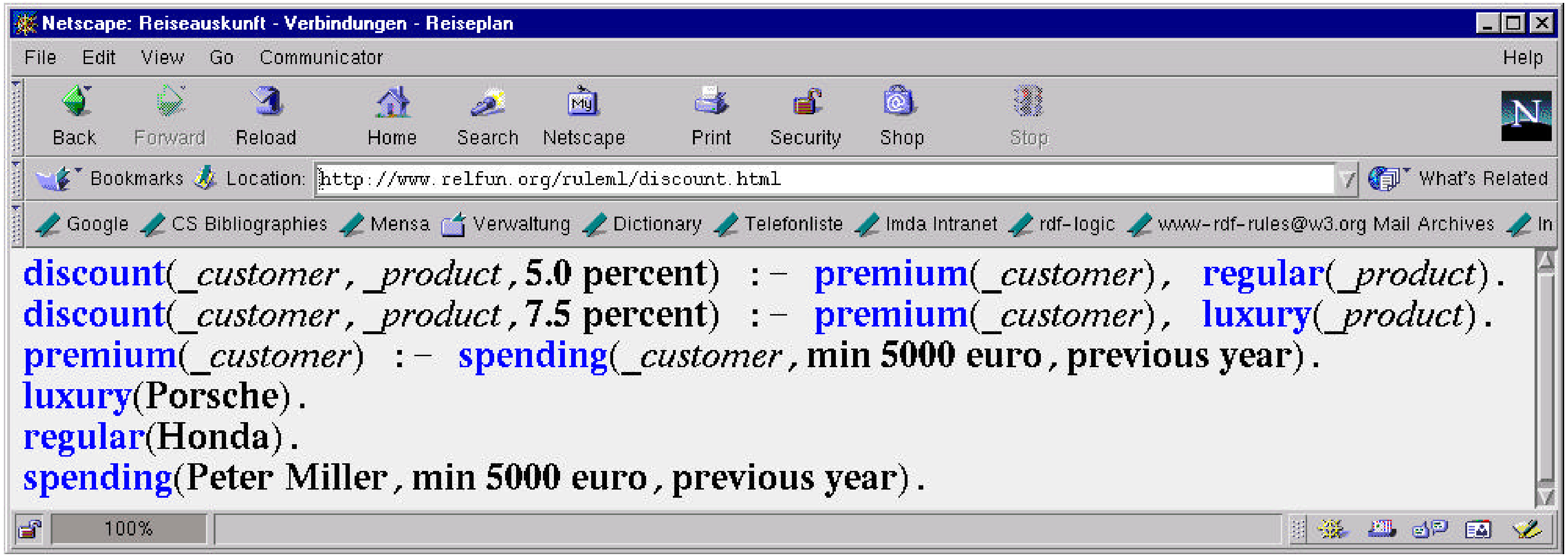
English Business Rules:

"The **discount** for a *customer* buying a *product* is **5.0 percent** if the *customer* is **premium** and the *product* is **regular**."

"The **discount** for a *customer* buying a *product* is **7.5 percent** if the *customer* is **premium** and the *product* is **luxury**."

. . .

Prolog-like formalization (syntax generated from XML):

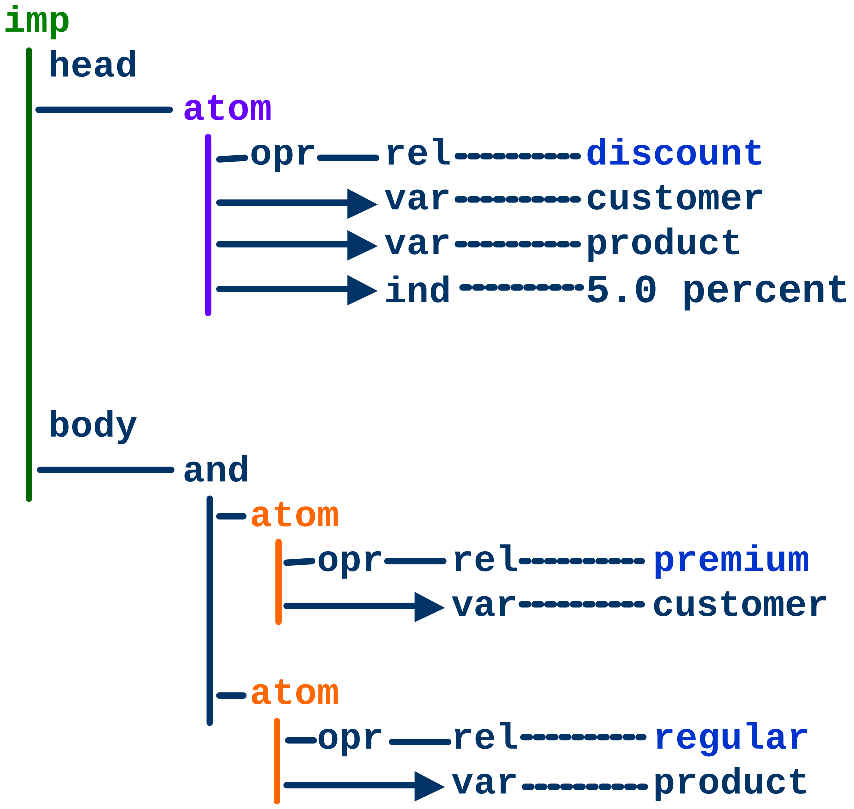


```
discount(_customer,_product,5.0 percent)  :-  premium(_customer),  regular(_product).
discount(_customer,_product,7.5 percent)  :-  premium(_customer),  luxury(_product).
premium(_customer)  :-  spending(_customer,min 5000 euro,previous year).
luxury(Porsche).
regular(Honda).
spending(Peter Miller,min 5000 euro,previous year).
```

XML-RDF RuleML 0.8: Markup and Tree

"The **discount** for a *customer* buying a *product* is **5.0 percent**
if the *customer* is **premium** and the *product* is **regular**."

```
<imp>
  <_head>
    <atom>
      <_opr><rel>discount</rel></_opr>
      <var>customer</var>
      <var>product</var>
      <ind>5.0 percent</ind>
    </atom>
  </_head>
  <_body>
    <and>
      <atom>
        <_opr><rel>premium</rel></_opr>
        <var>customer</var>
      </atom>
      <atom>
        <_opr><rel>regular</rel></_opr>
        <var>product</var>
      </atom>
    </and>
  </_body>
</imp>
```



RDF RuleML 0.8: Principles

- Use abbreviated ‘type - property’-alternating (“striped”) RDF syntax (similar to nested property lists), which nests subtrees and employs types as `rdf:Descriptions`:
  - A particular rule base becomes a (normally anonymous) RDF resource of type `rulebase` with a `_clauses` property/role leading to its `rdf:Seq`-type of rules labeled `rdf:li` for `rdf:_1`, `rdf:_2`, ...
  - An `imp` rule has `_head` and `_body` properties/roles leading to type-`atom` or type-`and` resources
  - Etc., down to RuleML's PCDATA leaves for relation symbols, individual constants, and variables, which become corresponding resources with `ruleml:cdata` literals in RDF

RDF RuleML 0.8: Striped Serialization

```
<rdf:RDF xmlns:rdf="&rdf;" xmlns:ruleml="&ruleml;" xmlns="&ruleml;">

<rulebase>            type
 <_clauses>            role
  <rdf:Seq>             type
   <rdf:li>              role
     <imp>                . . .
       <_head>
         <atom>
           <_opr><rel ruleml:cdata="discount"/></_opr>
           <_arg>
             <rdf:Seq>
              <rdf:li><var ruleml:cdata="customer"/></rdf:li>
              <rdf:li><var ruleml:cdata="product"/></rdf:li>
              <rdf:li><ind ruleml:cdata="5.0 percent"/></rdf:li>
             </rdf:Seq>
           </_arg>
         </atom>
       </_head>
       <_body>
         <and>
          <_arg>
           <rdf:Seq>
            <rdf:li>
             <atom>
              <_opr><rel ruleml:cdata="premium"/></_opr>
              <_arg>
               <rdf:Seq>
                <rdf:li>
                 <var ruleml:cdata="customer"/>
                </rdf:li>
               </rdf:Seq>
              </_arg>
             </atom>
            </rdf:li>
            <rdf:li>
             <atom>
              <_opr><rel ruleml:cdata="regular"/></_opr>
              <_arg>
               <rdf:Seq>
                <rdf:li>
                 <var ruleml:cdata="product"/>
                </rdf:li>
               </rdf:Seq>
              </_arg>
             </atom>
            </rdf:li>
           </rdf:Seq>
          </_arg>
         </and>
        </_body>
      </imp>
    </rdf:li>
   </rdf:Seq>
  </_clauses>
 </rulebase>
</rdf:RDF>
```

Original online

RDF RuleML 0.8: N-Triples Format

```
_:j17476 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#rulebase> .
_:j17477 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq> .
_:j17478 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#imp> .
_:j17479 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#atom> .
_:j17480 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#rel> .
_:j17480 <http://www.ruleml.org/rdf#cdata> "discount" .
_:j17479 <http://www.ruleml.org/rdf#_opr> _:j17480 .
_:j17481 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq> .
_:j17482 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#var> .
_:j17482 <http://www.ruleml.org/rdf#cdata> "customer" .
_:j17481 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> _:j17482 .
_:j17483 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#var> .
_:j17483 <http://www.ruleml.org/rdf#cdata> "product" .
_:j17481 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_2> _:j17483 .
_:j17484 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#ind> .
_:j17484 <http://www.ruleml.org/rdf#cdata> "5.0 percent" .
_:j17481 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_3> _:j17484 .
_:j17479 <http://www.ruleml.org/rdf#_arg> _:j17481 .
_:j17478 <http://www.ruleml.org/rdf#_head> _:j17479 .
_:j17485 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#and> .
_:j17486 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq> .
_:j17487 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#atom> .
_:j17488 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#rel> .
_:j17488 <http://www.ruleml.org/rdf#cdata> "premium" .
_:j17487 <http://www.ruleml.org/rdf#_opr> _:j17488 .
_:j17489 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq> .
_:j17490 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#var> .
_:j17490 <http://www.ruleml.org/rdf#cdata> "customer" .
_:j17489 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> _:j17490 .
_:j17487 <http://www.ruleml.org/rdf#_arg> _:j17489 .
_:j17486 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> _:j17487 .
_:j17491 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#atom> .
_:j17492 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#rel> .
_:j17492 <http://www.ruleml.org/rdf#cdata> "regular" .
_:j17491 <http://www.ruleml.org/rdf#_opr> _:j17492 .
_:j17493 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq> .
_:j17494 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ruleml.org/rdf#var> .
_:j17494 <http://www.ruleml.org/rdf#cdata> "product" .
_:j17493 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> _:j17494 .
_:j17491 <http://www.ruleml.org/rdf#_arg> _:j17493 .
_:j17486 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_2> _:j17491 .
_:j17485 <http://www.ruleml.org/rdf#_arg> _:j17486 .
_:j17478 <http://www.ruleml.org/rdf#_body> _:j17485 .
_:j17477 <http://www.w3.org/1999/02/22-rdf-syntax-ns#_1> _:j17478 .
_:j17476 <http://www.ruleml.org/rdf#_clauses> _:j17477 .
```

RDF RuleML 0.8: GraphViz Shape

Since this doesn’t show any details,
click the [Original online]

RDFS Types

Overview of RDFS Taxonomies for RuleML

- RuleML 0.8 still uses an unsorted logic, although this can simulate typed/sorted variables by applying distinguished unary predicates to those variables
- Moreover, with a special treatment of sorts and sorted variables, proofs can be kept at a more abstract level, thus reducing the search space
- A sort hierarchy is definable independently as the taxonomy of an Order-Sorted Logic or Description Logic, and be notated in RDFS, DAML+OIL, or OWL
- We are discussing preliminary constructs to link RuleML predicates/variables to externally defined RDFS classes (a similar mechanism is usable for ‘built-in’ XML datatypes)

How Typed RuleML Variables Can Link to RDFS / DAML+OIL / OWL Classes

- RuleML and Order-Sorted Logic or Description Logic class hierarchies – e.g. in RDFS, DAML+OIL, or OWL – go together well (RDFS, … properties will be harder)

- ‘Lift’ RDF’s use of `rdf:type` for taxonomic RDFS typing of **individuals**/resources (also for RuleML’s `inds`)

- New RDFS use: Access unchanged RDFS for typing of RuleML **variables**

“Type by Application” Technique

- In RuleML's conjunctive rule-body tag `and`
- give a taxonomic RDFS type to a logic variable
- by applying an RDFS class via a `rel`
  - containing the RDF attribute `rdf:resource`
  - to that logic variable

A Discounting Rule with Customer and Product Variables Typed by Applications:

```
<imp>
 <_head>
  <atom>
   <_opr><rel>discount</rel></_opr>
   <var>cust</var>        <!-- typed as Customer, see below -->
   <var>prod</var>        <!-- typed as Product, see below -->
   <ind>5.0 percent</ind>
  </atom>
 </_head>
 <_body>
  <and>
   <atom>
    <_opr><rel rdf:resource="http://description.org/ebiz#Customer"/></_opr>
    <var>cust</var>
   </atom>
   <atom>
    <_opr><rel>premium</rel></_opr>
    <var>cust</var>        <!-- typed as Customer, see above -->
   </atom>
   <atom>
    <_opr><rel rdf:resource="http://description.org/ebiz#Product"/></_opr>
    <var>prod</var>
   </atom>
   <atom>
    <_opr><rel>regular</rel></_opr>
    <var>prod</var>        <!-- typed as Product, see above -->
   </atom>
  </and>
 </_body>
</imp>
```

Given that
cust has type Customer
and
prod has type Product,

the discount for
a **cust**
buying
a **prod**
is 5.0 percent

if

the **cust** is premium
and
the **prod** is regular.

“Type by Declaration” Technique

- In RuleML's Horn-clause tags `fact` and `imp`
- give a taxonomic RDFS type to a logic variable
- by referring to an RDFS class via an `rdf:type`-like
  - RuleML role `_type`
  - containing the RDF attribute `rdf:resource`

A Discounting Rule with Customer and Product Variables Typed by Declarations

```
<imp>
 <_type rdf:resource="http://description.org/ebiz#Customer"> <var>cust</var> </_type>
 <_type rdf:resource="http://description.org/ebiz#Product"> <var>prod</var> </_type>
 <_head>
   <atom>
     <_opr><rel>discount</rel></_opr>
     <var>cust</var>        <!-- typed as Customer, see above -->
     <var>prod</var>        <!-- typed as Product, see above -->
     <ind>5.0 percent</ind>
   </atom>
 </_head>
 <_body>
   <and>
     <atom>
       <_opr><rel>premium</rel></_opr>
       <var>cust</var>        <!-- typed as Customer, see above -->
     </atom>
     <atom>
       <_opr><rel>regular</rel></_opr>
       <var>prod</var>        <!-- typed as Product, see above -->
     </atom>
   </and>
 </_body>
</imp>
```

Given that
cust has type Customer
and
prod has type Product,

the discount for
a **cust**
buying
a **prod**
is 5.0 percent

if

the **cust** is premium
and
the **prod** is regular.

Typing Scope and Multiple Typing

- Reflecting the scope of logic variables – which is a single clause (`fact` or `imp`) – the typing scope is the clause containing the `rel` application or the `_type` role
- To express RDF-like multiple (intersection) types, just use these multiple types for one logic variable, e.g. European ? Customer would become:

```
<atom>
  <_opr><rel rdf:resource="http://description.org/ebiz#Customer"/></_opr>
  <var>cust</var>
</atom>
<atom>
  <_opr><rel rdf:resource="http://description.org/ebiz#European"/></_opr>
  <var>cust</var>
</atom>
```

or

```
<_type rdf:resource="http://description.org/ebiz#Customer"> <var>cust</var> </_type>
<_type rdf:resource="http://description.org/ebiz#European"> <var>cust</var> </_type>
```

Types, Description Logics, and Ontologies

- Order-Sorted Horn logics have provided a solid foundation for implementing such hierarchical types, possibly employing
  - a DL-like classifier during unification or even
  - a corresponding mechanism during indexing
- Summary:
  - Such RDFS-RuleML links begin to realize the ‘equation’ **Ontology = Taxonomy + Rules**

RDFS Types

References

- Harold Boley: Relationships Between Logic Programming and RDF, in: R. Kowalczyk, S.W. Loke, N.E. Reed, G. Graham (Eds.), Advances in Artificial Intelligence, LNAI 2112, Springer-Verlag, 2001
- Harold Boley: A Web Data Model Unifying XML and RDF. Draft, September 2001.
- Harold Boley: The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations, Invited Talk, INAP2001, Tokyo, October 2001.
- Harold Boley, Said Tabet, and Gerd Wagner: Design Rationale of RuleML: A Markup Language for Semantic Web Rules, Proc. SWWS'01, Stanford, July/August 2001.
- Andreas Eberhart, An Agent Infrastructure based on Semantic Web Standards, Workshop on Business Agents and the Semantic Web at the AI 2002, Calgary, Canada
- Andreas Eberhart, Automatic Generation of Java/SQL based Inference Engines from RDF Schema and RuleML, International Semantic Web Conference 2002, Sardinia
- Benjamin Grosof: Representing E-Business Rules for the Semantic Web: Situated Courteous Logic Programs in RuleML, Proc. Workshop on Information Technologies and Systems (WITS '01), New Orleans, December, 2001.
- Benjamin Grosof, Mahesh D. Gandhe, and Timothy W. Finin: SweetJess: Translating DamlRuleML to Jess, Proc. International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, Sardinia (Italy), June 2002.
- Benjamin Grosof and Terrence Poon: Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions, Proc. International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, Sardinia (Italy), June 2002.
- Steve Ross-Talbot, Harold Boley, and Said Tabet: Playing by the Rules, Application Development Advisor 6(5), June 2002, 38-43.
- Michael Schroeder and Gerd Wagner (Eds.): Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web. Sardinia, Italy, June 14, 2002. CEUR-WS Publication Vol-60.
- Gerd Wagner: How to Design a General Rule Markup Language?, Invited Talk, Workshop XML Technologien für das Semantic Web (XSW 2002), Berlin, June 2002.

Refs