The Rule Markup Initiative: Syntax Examples and DTD's incl. Modularization

DTDs

Harold Boley
Benjamin Grosof
Said Tabet

February 25, 2001

Initial Example: Backward-Rule Notation

Further formalized RuleML example (still unanalyzed English relation and individual-constant names):

```
<if>
  <atom>
    <rel>may look at</rel>
    <var>you</var>
    <ur label="Rule-Based Systems">http://www.cs.brandeis.edu/...</ur>
  </atom>
  <atom>
    <rel>want to review</rel>
    <var>you</var>
    <ind>rule principles</ind>
  </atom>
</if>
```

conclusion

premise

Clocksin/Mellish Sample Prolog Clauses

Rule (Non-unit clause):

```
<if>
  <atom>
    <rel>likes</rel>      likes(
    <ind>John</ind>             john,
    <var>x</var>                X )
  </atom>               :-
  <atom>
    <rel>likes</rel>      likes(
    <var>x</var>                X,
    <ind>wine</ind>             wine ) .
  </atom>
</if>
```

Fact (Unit clause):

```
<if>
  <atom>
    <rel>likes</rel>
    <ind>Mary</ind>
    <ind>wine</ind>
  </atom>
  <and/>
</if>
```

Empty ‘and’ → true premise → *fact*ual rule

likes(mary,wine).

Proposed W3C-Page Authentication Rule

Tim Berners-Lee: *Any person who was some time in the last 2 months an employee of an organization which was some time in the last 2 months a W3C member may register*

```
<if>
 <atom>
  <rel>may register</rel>
  <var>x</var>
 </atom>
 <and>
  <atom>
   <rel>person</rel>
   <var>x</var>
  </atom>
  <atom>
   <rel>organization</rel>
   <var>y</var>
  </atom> ...
```


Authentication Rule continued

Tim Berners-Lee: *Any person who was some time in the last 2 months an employee of an organization which was some time in the last 2 months a W3C member may register*

```
<atom>
  <rel>employee in</rel>
  <var>x</var>
  <var>y</var>
  <cterm>
   <ctor>last</ctor>
   <cterm>
    <ctor>month</ctor>
    <ind>2</ind>
   </cterm>
  </cterm>
</atom>
...
```


Authentication Rule continued more

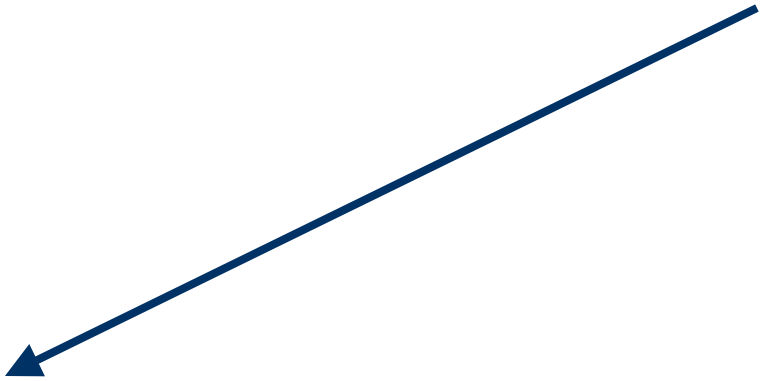
Tim Berners-Lee: *Any person who was some time in the last 2 months an employee of an organization which was some time in the last 2 months a W3C member may register*

```
<atom>
    <rel>member in</rel>
    <var>org</var>
    <ur label="W3C">http://
            www.w3.org/</ur>
    <cterm>
     <ctor>last</ctor>
     <cterm>
      <ctor>month</ctor>
      <ind>2</ind>
     </cterm>
    </cterm>
  </atom>
</and>
</if>
```


Directed Equations (Rewriting) -- Example: use them for URI Expansion

uriexp(daml) := http://www.daml.org/

URLs/URIs or URs
as 1st-class citizens



```
<if>
 <eq>
  <nano>
   <fun>uriexp</fun>
   <ind>daml</ind>
  </nano>
  <ur>http://www.daml.org/</ur>
 </eq>
 <and/>
</if>
```

. . .

RDF Triples as Very Special Rules

RDF triple (predicate, subject, object) as triple (rel, ur, ur/ind):

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

(Creator, http://www.w3.org/Home/Lassila, Ora Lassila)

```
<if>
  <atom>
    <rel>Creator</rel>
    <ur>http://www.w3.org/Home/Lassila</ur>
    <ind>Ora Lassila</ind>
  </atom>
  <and/>
</if>
```

Modularization of DTDs: XHTML and KR

Advantages:

- Leads to reusable subDTDs and DTD interoperation
- Complex DTDs built with 'plug-and-play' technology
- (RuleML) Sublanguages determined by validations!
- For (rulebase) export find most precise sublanguage!

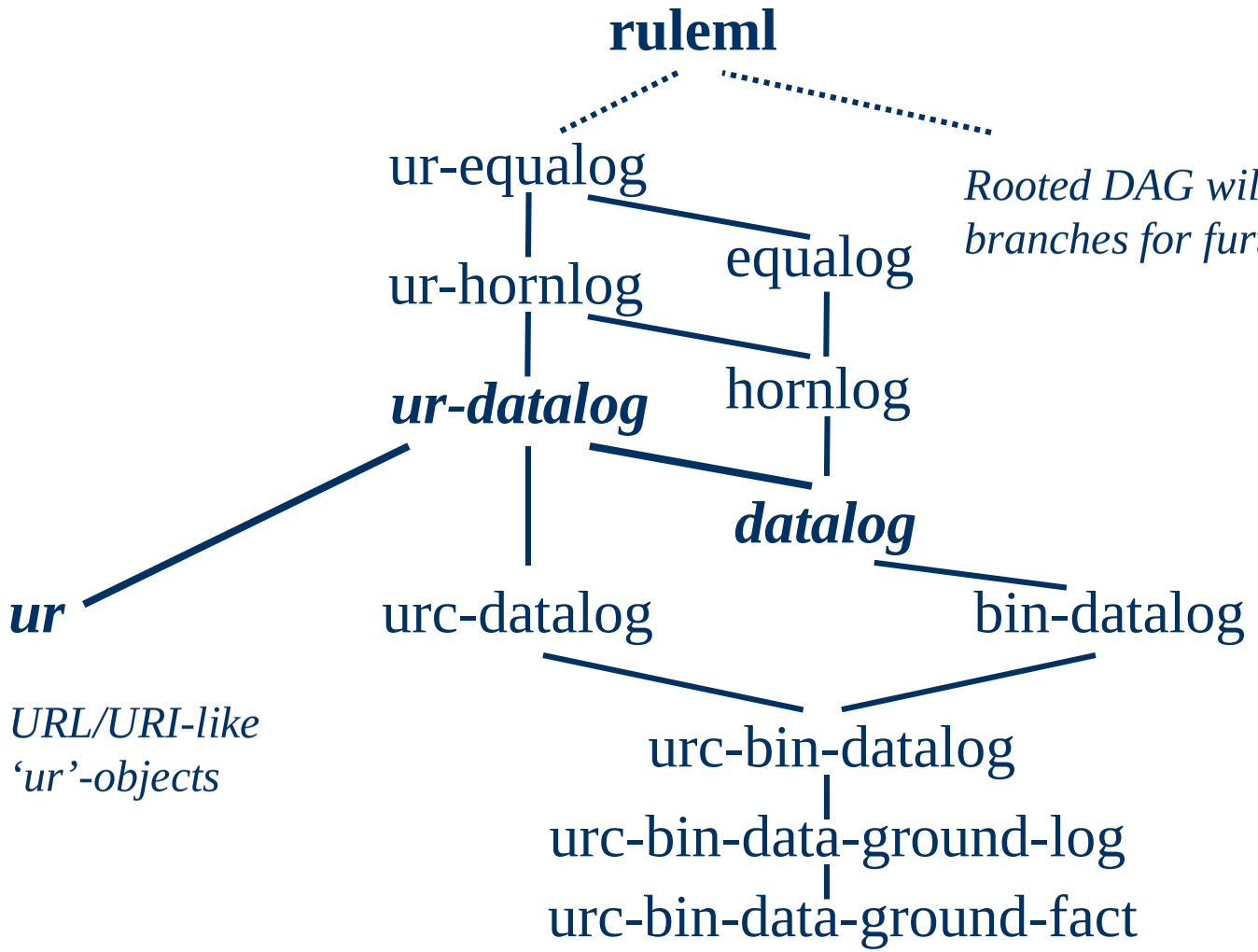
Modular DTDs still mostly used outside KR:

- First used for XHTML and described in XML Bible
- W3C Working Draft 5 January 2000 Building XHTML[tm] Modules
- W3C Candidate Recommendation 20 October 2000 Modularization of XHTML[tm]

Structure of the RuleML DTD Hierarchy

- Our system of DTDs (current version: 0.7) 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 corresponds to a specific RuleML sublanguage:
  - '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



Rooted DAG will be extended with branches for further sublanguages

ur-datalog = join(ur,datalog)

URL/URI-like 'ur'-objects

RDF-like triples

A Relational Language: ruleml-datalog.dtd (I)

```
<!-- An XML DTD for a Datalog RuleML Sublanguage -->
<!-- Last Modification: 2001-01-25 -->

<!-- ENTITY Declarations -->

<!-- in this ruleml-datalog.dtd, parameter entities set two *.module switches to INCLUDE -->

<!ENTITY % datalog.module "INCLUDE">
<!ENTITY % datalog-and-hornlog.module "INCLUDE">

<!-- hence all conditional sections "<![%*.module;[" . . . "]]>" activate their content; -->
<!-- in a stand-alone use of the current DTD "<![%*.module;[" and "]]>" are thus no-ops -->

<![%datalog-and-hornlog.module;[

<!-- a conclusion and premise will be usable within 'if' implications -->
<!-- in datalog and hornlog, conc element uses an atomic formula -->
<!-- in datalog and hornlog, prem element uses an atomic formula or an 'and' -->

<!ENTITY % conc "atom">
<!ENTITY % prem "(atom | and)">

]]>
```

A Relational Language: ruleml-datalog.dtd (II)

```
<!-- ELEMENT and ATTLIST Declarations -->

<!-- 'rulebase' root element uses 'if' implications as top-level rules -->

<!-- label attribute allows naming of an entire individual rulebase; -->
<!-- e.g., this can help enable forward inferencing of selected rulebase(s)  -->

<!-- direction attribute indicates the intended direction of rule inferencing;  -->
<!-- it is a preliminary design choice and has a 'neutral' default value -->

<!ELEMENT rulebase (if*)>
<!ATTLIST rulebase label CDATA #IMPLIED>
<!ATTLIST rulebase direction (forward | backward | bidirectional) "bidirectional">


<!-- 'if' implications are usable as top-level rules -->
<!-- 'if' element uses a conclusion followed by a premise -->
<!-- "<if>conc prem</if>" stands for "conc if prem", i.e., "conc is true if prem is true" -->

<!-- label attribute is a handle for the rule: for various uses, including editing -->

<!ELEMENT if (%conc;, %prem;)>
<!ATTLIST if label CDATA #IMPLIED>
```


A Relational Language: ruleml-datalog.dtd (III)

```
<![%datalog-and-hornlog.module;[

<!-- an 'and' is usable within premises -->
<!-- 'and' uses zero or more atomic formulas -->
<!-- "<and>atom</and>" is equivalent to "atom"-->
<!-- "<and></and>" is equivalent to "true"-->

<!ELEMENT and (atom*)>

]]>


<![%datalog.module;[

<!-- atomic formulas are usable within conc's, prem's, and 'and's -->
<!-- atom element uses rel(ation) symbol followed by a sequence of -->
<!-- zero or more arguments, which may be ind(ividual)s or var(iable)s -->

<!ELEMENT atom (rel, (ind | var)*)>

]]>
```

A Relational Language: ruleml-datalog.dtd (IV)

<!-- there is one kind of fixed argument -->

<!-- individual constant, as in predicate logic -->

<!ELEMENT ind (#PCDATA)>

<!-- there is one kind of variable argument -->

<!-- logical variable, as in logic programming -->

<!ELEMENT var (#PCDATA)>

<!-- there are only fixed (first-order) relations -->

<!-- relation or predicate symbol -->

<!ELEMENT rel (#PCDATA)>

A URL/URI Language: ruleml-ur.dtd

```
<!-- An XML DTD for a 'UR' RuleML Sublanguage -->
<!-- Last Modification: 2001-01-23 -->


<!-- ENTITY Declarations -->

<!-- a Uniform Resource Identifier is currently PCDATA, but see W3C's [RFC2396] -->

<!ENTITY % URI "#PCDATA">


<!-- ELEMENT and ATTLIST Declarations -->

<!-- there is an additional kind of fixed argument -->

<!-- objects (resources) use a URL/URI as their OID, as in SHOE or RDF (cf. URML) -->
<!-- however, unlike for XHTML anchors etc. URI used as content, not as attribute -->
<!-- 'label' attribute, unlike URI, need not be unique -->
<!-- if no 'label' attribute is given, browser must highlight the URI itself -->

<!ELEMENT ur (%URI;)>
<!ATTLIST ur label CDATA #IMPLIED>
```


The Join Language: ruleml-urdatalog.dtd

```
<!-- An XML DTD for a 'UR' Datalog RuleML Sublanguage -->
<!-- Last Modification: 2001-01-23 -->

<!-- ENTITY Declarations -->

<!ENTITY % urdatalog.module "INCLUDE">
<!ENTITY % datalog.module "IGNORE">

<!ENTITY % datalog SYSTEM "ruleml-datalog.dtd">
%datalog;

<!ENTITY % ur SYSTEM "ruleml-ur.dtd">
%ur;

<!-- ELEMENT and ATTLIST Declarations -->

<![%urdatalog.module;[

<!-- atomic formulas are usable within conc's, prem's, 'and's, and 'or's -->
<!-- atom element uses rel name followed by three kinds of arguments -->

<!ELEMENT atom (rel, (ur | ind | var)*)>

]]>
```

Conclusions

- RuleML DTD 0.7, a system of 12 DTDs, is available at http://www.dfki.de/ruleml/indtd.html
- Sample files -- each referring to the most specific DTD still validating them -- are at http://www.dfki.de/ruleml/exa
- Further rule categories (e.g. ICs and triggers) and DTD updates will be available via main RuleML page at http://www.dfki.de/ruleml
- Distributed KR can already be based on current DTDs -- using (XSLT) transformations to reach follow-up and participants' DTDs

