

FOL RuleML: Release, Use, and Outlook

Harold Boley, Presenter
NRC IIT e-Business

Joint Work with
Benjamin Grosf and Said Tabet
as part of the
RuleML Initiative and Joint Committee

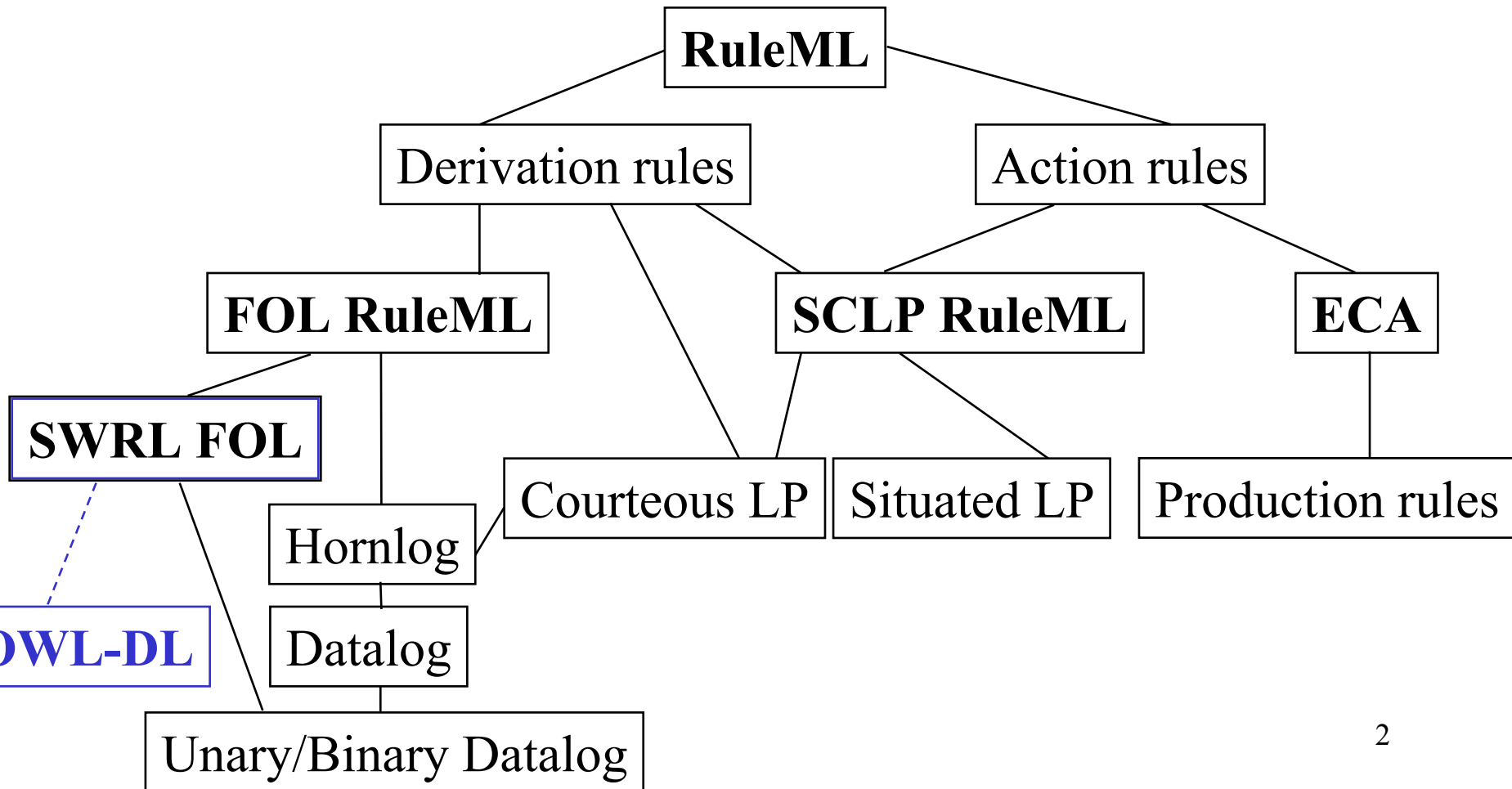
DAML PI Meeting, San Antonio, TX
Nov 30 – Dec 02, 2004



Outline

- RuleML V0.87
 - Slots generalized for F-Logic
- FOL RuleML V0.9
 - Release; Synergy with SWRL FOL
- DTD and Example
- Applications at NRC; OMG and RuleML
- Plan for full RuleML V0.9
 - Tighter SWRL convergence
 - Include-a-KB (refine SweetRules design)
 - Action/reaction rules

SWRL FOL and (FOL/SCLP) RuleML



RuleML 0.87

- Groundwork for FOL (First-Order-Logic) RuleML
 - Markup economy as in RuleML Lite: stripe skipping
- Access to [SWRL properties as “foreign” atoms](#)
- [UML for language lattice](#), [MOF for abstract syntax](#)
- Strong negation (Neg) & Negation-as-failure (Naf)
 - More support for Courteous LP (prev.: rule labels)
- Slotted syntax permits (generalized) attribute sets
 - In addition to positional syntax
 - Facilitates N3 / RDF / OWL / SWRL style
 - (Variable, Complex) terms as slot names: for F-Logic

FOL RuleML 0.9

- FOL RuleML 0.9 announced: [2004-11-14](#)
- Co-developed as platform for Web rules by RuleML Initiative and Joint Committee
- Roles of FOL RuleML:
 - the FOL *sublanguage* of **RuleML**
 - extends rule *component* of **SWRL FOL**
 - an FOL *content language* for **SWSI**

(FOL) RuleML Has N-ary Relations & Functions, Extending SWRL (FOL)

- N-ary relations (predicate symbols)
 - Extends SWRL, which is unary/binary
- N-ary constructors (function symbols)
 - Extends SWRL, which uses individuals as 0-ary constructors (function-free)

FOL RuleML: Syntax and Semantics

- Modular combination of syntactically characterized sublanguages with:
 - Explicit quantifiers (also: LP convention)
 - Head disjunctions
 - Equivalence and Negation
- Semantics is FOL model theory
- (Pragmatics via performatives)

Example

FOL RuleML:

English:
 “If
 a person buys an object from a merchant
 and
 the person keeps the object
 then
 the person owns the object.”

```

<Forall>
  <Var>person</Var>
  <Var>merchant</Var>
  <Var>object</Var>
  <Implies>
    <And>
      <Atom>
        <Rel>buy</Rel><Var>person</Var><Var>merchant</Var><Var>object</Var>
      </Atom>
      <Atom>
        <Rel>keep</Rel><Var>person</Var><Var>object</Var>
      </Atom>
    </And>
    <Atom>
      <Rel>own</Rel><Var>person</Var><Var>object</Var>
    </Atom>
  </Implies>
</Forall>
  
```


FOL (binary/ternary, function-free):
 $(\forall \text{ person, merchant, object})$
 $\text{buy}(\text{person, merchant, object})$
 \wedge
 $\text{keep}(\text{person, object})$
 \rightarrow
 $\text{own}(\text{person, object})$ 7

DTD for Recursive FO Formulas

```
<!ENTITY % foformula
    "(Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)">
<!ELEMENT Atom (Rel, (Ind | Var | Cterm)*)>
<!ELEMENT Cterm (Ctor, (Ind | Var | Cterm)*)>
<!ELEMENT And ((%foformula;)*)>
<!ELEMENT Or ((%foformula;)*)>
<!ELEMENT Neg (%foformula;)>
<!ELEMENT Implies (%foformula;, %foformula;)>
<!ELEMENT Equivalent (%foformula;, %foformula;)>
<!ELEMENT Forall (Var+, %foformula;)>
<!ELEMENT Exists (Var+, %foformula;)>
<!ELEMENT Ind (#PCDATA)>
<!ELEMENT Var (#PCDATA)>
<!ELEMENT Rel (#PCDATA)>
<!ELEMENT Ctor (#PCDATA)>
```

☞ *Translated to XML Schema
for SWRL FOL spec ⁸*

Slotted FOL RuleML Extension

- N-ary relations and constructors can contain set of slots (“name→filler” pairs)
 - Enables Object Oriented modeling
 - RDF URI descriptions (rather than triples)
 - RDFS and OWL class descriptions
 - Positional logic  Frame logic (F-logic)
- Markup for F-logic in OO RuleML

Example (Original)

Positional FOL RuleML:

```

<Forall>
  <Var>person</Var>
  <Var>merchant</Var>
  <Var>object</Var>
  <Implies>
    <And>
      <Atom>
        <Rel>buy</Rel><Var>person</Var><Var>merchant</Var><Var>object</Var>
      </Atom>
      <Atom>
        <Rel>keep</Rel><Var>person</Var><Var>object</Var>
      </Atom>
    </And>
    <Atom>
      <Rel>own</Rel><Var>person</Var><Var>object</Var>
    </Atom>
  </Implies>
</Forall>
  
```

Positions

arg 1

arg 2

arg 3

Example (Extended)

Slotted FOL RuleML:

```

<Forall>
  <Var>person</Var>
  <Var>merchant</Var>
  <Var>object</Var>
  <Implies>
    <And>
      <Atom>
        <Rel>buy</Rel><Var>person</Var><Var>merchant</Var><Var>object</Var>
      </Atom>
      <Atom>
        <Rel>keep</Rel><Var>person</Var><Var>object</Var><Slot><Ind>Δt</Ind><Ind>'04</Ind></Slot>
      </Atom>
    </And>
    <Atom>
      <Rel>own</Rel><Var>person</Var><Var>object</Var><Slot><Ind>Δt</Ind><Ind>'04</Ind></Slot>
    </Atom>
  </Implies>
</Forall>

```

Slot

name → filler

Exemplary RuleML Apps: NRC

- RACSA, RALOCA, RACOFI: Rule Applying Agents for Comparison Shopping, Learning Object Comparison, and COllaborative FIltering (commercial: inDiscover.net)
- [NBBizKB](#): New Brunswick Business Knowledge Base uses OO RuleML for data validation and [integration](#)
- [AgentMatcher](#): e-Learning metadata interchanged in Weighted OO RuleML
- [Teclantic](#): Profiles of startup companies for Atlantic technology transfer in Weighted OO RuleML

OMG and RuleML

- OMG has Production & Business Rule RFPs
 - Focus: meta-model of OMG
 - Considering RuleML for markup and KR semantics
- RuleML developed UML, MOF & MDA specs
 - [UML for language lattice](#) (OCL planned)
 - [MOF for abstract syntax](#)
 - MDA: platform-independent business logic¹³

Plan '05 for RuleML 0.9: Toplevel

- Transfer the FOL RuleML 0.9 innovations to all sublanguages, esp. LP
- Tighten SWRL convergence (e.g., 1-, 2-, n-ary)
- Finalize RuleML/SWRL datatypes and built-ins
- Update equality (active: rewrite/transformation)
- Permit (FOL) integrity constraints: Mutex, OCL
- Develop webized, KB-converting 'Includes'
- Extend RuleML towards action/reaction rules
 - Production rules / Situated Courteous LP