The RuleML Knowledge-Interoperation Hub

Harold Boley

Faculty of Computer Science, University of New Brunswick, Canada

The 10th International Web Rule Symposium, RuleML 2016
Stony Brook University, 6-9 July 2016

Interoperation via Representation and Transformation

- All **hub** & **spoke** **knowledge interoperation** uses **canonical representation language**, with **translators** **mapping in and out**
- Web-based RuleML tools for *interoperation* (representation and transformation) reached critical mass, with **novel** **translator** **chains** mapping through **RuleML/XML**

- Representation:
  Syntactic and Semantic
- Transformation:
  Internal and External
- Interoperation:
  N3-PSOA-Flora Use Case

RuleML Logic as Syntactic Language with Semantic Profile

- RuleML is a system of families of **languages** of XML-serialized instance documents (containing KBs and queries) specified syntactically through schemas and optionally associated with semantic **profiles** through syntax-semantics-pairing **logics**
- For each pair *logic* = (*language*, *profile*), *language* is predefined but *profile* and *logic* are predefined or user-defined (where *logic* can be predefined only if *profile* is)

Configuring Textbook and Enriched Relax NG Syntax

- RuleML's modular Relax NG schemas permit rule interchange with high precision
- MYNG accepts set of desired RuleML language features and configures Relax NG schema for that language
- Allows supplementary (Semantic) Web language features such as IRIs, OIDs, types, and slots, e.g. in Datalog($^+$) and Hornlog($^+$)
- RuleML is also introducing "textBooK" (BK) versions without supplementary features, e.g. DatalogBK($^+$) and HornlogBK($^+$)

Semantic Profiles: Two Sample Descriptors

1. Top-Down Classification:
   - Proof(-theoretic): Resolution vs. ASP etc.
   - Model(-theoretic): Herbrand vs. Tarski
2. Reference to Web-published semantics and mapping between *its* syntax and RuleML/XML syntax

Knowledge Transformation Tool Suite: Basics

Implemented semantics-preserving translators

Terminology:

- RuleML/XML is the ‘machine-oriented’ RuleML serialization syntax
- *RuleML/short* stands for ‘human-oriented’ RuleML shorthand syntaxes such as POSL and PSOA/PS
- *foreign* stands for non-RuleML syntaxes such as Prolog and RIF/PS

Knowledge Transformation Tool Suite: Taxonomy

- Internal: RuleML-to-RuleML
  - Serialized: RuleML/XML-to-RuleML/XML
    - Upgraders (e.g., to Version 1.02)
    - Formatters (e.g., for Version 1.02)
    - Normalizer (Section 3.1)
    - Compactifiers (Section 3.1)
  - Polarized (Section 3.2): Between-RuleML/XML-and-*RuleML/short*
    - Parsers: *RuleML/short*-to-RuleML/XML
    - Generators: RuleML/XML-to-*RuleML/short*
- External: Between-RuleML/XML-and-*foreign*
  - Importers (Section 3.3): *foreign*-to-RuleML/XML
  - Exporters (Section 3.3): RuleML/XML-to-*foreign*

Transformation Chaining

- Parsers and Generators under Polarized sub-branch of Internal branch as well as Importers and Exporters of External branch can be composed
- Creates transformation chains mapping through RuleML/XML as in following compositions, where POSL and PSOA/PS are two 'shorthand' syntaxes for a Deliberation RuleML subset, while Dexlog and TPTP refer to subsets of two 'foreign' syntaxes:
  - Internal-Internal: POSL$\longrightarrow$RuleML/XML$\longrightarrow$PSOA/PS
  - External-External: Dexlog$\longrightarrow$RuleML/XML$\longrightarrow$TPTP
  - Internal-External: POSL$\longrightarrow$RuleML/XML$\longrightarrow$TPTP
  - External-Internal: Dexlog$\longrightarrow$RuleML/XML$\longrightarrow$PSOA/PS

N3-PSOA-Flora Interoperation Use Case: Challenge

- Bridging the gap between two languages for rule-based Semantic Web, also supporting (light-weight-)ontology-based Semantic Web: N3 and Flora-2/F-logic
- Interoperation from N3 to Flora-2/F-logic, although opposite direction can be analogously constructed from alignment
- Also shows PSOA RuleML as intermediate (canonical) format that focuses entirely on knowledge-representation layer rather than programming-language details, but makes syntactic assumptions (e.g. quantifiers) explicit

N3-PSOA-Flora Interoperation Use Case: Template

(Controlled) English: “If the relation addressRel holds between a name, a street, and a town, then there exists an object, addressObj, with a name slot and a place slot for which there exists an object, placeObj, with a street slot and a town slot.”

- This rule and a fact will be given as N3 source, Flora-2/F-logic target, and variants of PSOA RuleML canonical form

N3-PSOA-Flora Interoperation Use Case: Source

N3 fact and rule, where default namespace (N3's ":" prefix) is RuleML's GeospatialRules and `rel:arglist` is N3 property defined in PSOA RuleML namespace for N3 vocabulary that emulates relations:

```
@prefix : <http://psoa.ruleml.org/GeospatialRules#>.
@prefix rel: <http://psoa.ruleml.org/n3/vocab/rel#>.

[a :addressRel;
 rel:arglist ("Computer Science" "Engineering Dr"
                                 "Stony Brook, NY 11794")].
{
 [a :addressRel;
  rel:arglist (?Name ?Street ?Town)]
}
=>
{
 [a :addressObj;
  :name ?Name;
  :place [a :placeObj;
          :street ?Street;
          :town ?Town]]
}.
```

N3-PSOA-Flora Interoperation Use Case: Target

Flora-2/F-logic fact and rule, where compiler option for experts enables use of embedded ISA-literal (Flora-2's ":" infix) in rule head, as described in Flora-2 manual, Section 48:

```
:- compiler_options{expert=on}.

addressRel('Computer Science','Engineering Dr',
                                'Stony Brook, NY 11794').

\#(?Name,?Street,?Town):addressObj[
     name->?Name,
     place->\#(?Name,?Street,?Town):placeObj[
       street->?Street,
       town->?Town]] :-
    addressRel(?Name,?Street,?Town).
```

N3-PSOA-Flora Interoperation Use Case: Canonical

PSOA RuleML/PS fact and rule, where rule, from RW '15 Paper, uses FOL-style explicit quantifiers (adapted from FOL RuleML/XML as well as W3C RIF/XML and RIF/PS):

```
addressRel("Computer Science" "Engineering Dr"
                                   "Stony Brook, NY 11794")

Forall ?Name ?Street ?Town (
  Exists ?O1 ?O2 (
    ?O1#addressObj(name->?Name
                   place->?O2#placeObj(street->?Street
                                       town->?Town)) )  :-
  addressRel(?Name ?Street ?Town)
                                )
```

N3-PSOA-Flora Interoperation Use Case: Rule!

A) Rule can (1) be enriched by taxonomic subsumptions – using "##" infix – such as `addressObj##geoObj` and (2) be employed to align given facts populating *relational* address ontology such as above `addressRel` fact with derivable facts for populating *object-centered* address ontology such as following derivable fact:

```
skolem1#addressObj(
  name->"Computer Science"
  place->skolem2#placeObj(street->"Engineering Dr"
                          town->"Stony Brook, NY 11794"))
```

B) But such a rule is itself the subject of interoperation, by representation in XML and transformation with XSLT, as explored by the RuleML hub