RIF Design Roadmap

Draft 2006-02-27-6PM

Harold Boley (NRC), Michael Kifer (Stony Brook U),
Axel Polleres (DERI), Jos de Bruijn (DERI),
Michael Sintek (DFKI),
Giorgos Stamou (NTUA), Jeff Pan (U of Aberdeen)

Preliminary design roadmap aims at a consensual system

For RIF **Phase 1** features distilled from the current UCR draft, the RIFRAF, and the public archive

Builds on the Design Goals
(cf. http://www.w3.org/2005/rules/wg/wiki/UCR/Design_Goals)

Approach to RIF **Phase 2** also sketched

PHASE 1

1. Specify syntax and semantics of Horn Logic and sublanguages

(cf. http://www.ruleml.org/modularization/#Model)

1.1. Full Horn Logic (functions, no negation)

1.2. Datalog (Horn Logic without functions, no negation)

2. Syntactic and semantic extensions of Horn Logic …

2.1. Define purely syntactic extensions
(cf. http://www.w3.org/Submission/SWSF-SWSL)

2.1.1. Monotonic Lloyd-Topor extensions
(disjuncts in body, conjuncts in heads)

2.1.2. Named arguments (slots)
in n-ary notation
- Can be developed into frames
with OIDs (as in F-logic) in Phase 2

2.1.3. Higher-order syntax (cf. HiLog) → Phase 2

2.2. Support literals and datatypes
(common functions and operators)

… 2. Syntactic and semantic extensions of Horn Logic

2.3. Delineate appropriate semantics
for different Horn-like rules (cf. #9.2)
(could be moved to Phase 2)

2.3.1. First-order (all-model) semantics (cf. #5)

2.3.2. Minimal-model semantics

2.3.3. Semantics for production rules

3. Webizing features that should be (globally) addressable

3.1. Adopting IRIs (incl. URIs, URLs) for Web-based addressing

3.2. IRI addressing of RIF constants and predicates

3.3. IRI addressing of other RIF features

4. Pure production rules with only asserts in the action part

- Execution is 1-to-1 with model generation, semantics compatible with #1
- Basis for interoperation between production rules and Horn rules
- Action part will then be generalized in Phase 2

5. Integrity queries, a.k.a. integrity constraints

(will not require extra effort)

- These constraints are considered violated if the queries have answers (an answer is a witness to an integrity-constraint violation)
- The allowed queries must have the syntactic form of a rule body
- Semantics of a rule body as in #1. Pragmatics of a warning or an error

6. Scope feature for modularizing/ structuring rulebases

(cf. TRIPLE models/contexts, FLORA modules, named graphs)

- Will enable Load-and-Query rule engines (i.e., engines that can load and then query different rulebases at once)
- Units for tagging provenance etc. (cf. #9)
- Basis for scoped negation as failure in Phase 2

7. Interoperability with RDF

(work already ongoing)

- Treat an RDF graph as a ruleset of binary or ternary facts
- Treat blank nodes as existentials in rule bodies and as Skolem functions in rule heads
- Accommodate SPARQL queries from the body of rules (cf. #8)

8. Interoperability with OWL

(work already ongoing)

- Define a hybrid combination semantics

(cf. http://rewerse.net/deliverables/m12/i3-d3.pdf and ftp://ftp.cs.sunysb.edu/pub/TechReports/kifer/msa-ruleml05.pdf)

At this stage, interoperation between OWL and rules will be at the level of rule bodies posing ground (after instantiation) queries to OWL

9. Metadata/semantic attributes for rule documents, scopes, rules, facts …

(could be moved to Phase 2)

9.1. To enable searching for rulesets, Google Directory-style

9.2. To enable tagging rulesets with intended semantics (e.g., FOL, LP/well-founded). This may not be that important in Phase 1, but will certainly be in Phase 2 (cf. #2.3)

9. … Metadata/semantic attributes for rule documents, scopes, rules, facts

(could be moved to Phase 2)

9.3. To enable tagging rulesets to indicate syntactic features that should be supported by the recipient.
This will support conformance-guided rule system interoperation.
Tagging can be done by pointing to a suitable XML Schema document
(cf. #1)

10. XML Serialization

(cf. RuleML's serialization)

- RDF can be generated via XSLT

PHASE 2

Allow for three subfamilies with different semantics: FOL-style, LP-style, Production rules

The specific semantics will be indicated by a semantic attribute

Define different kinds of extended RDF/OWL interoperability for the three different kinds of extended rules

Optional extensions follow Phase 1 using the extensibility notion of the Charter

These extensions could be specified
in a separate RIF document
working horizontally across
the three subfamilies below

I. FOL-style rules

1. Based on Phase 1
2. Continued RDF Interoperability
3. SWRL-inspired rule extension of OWL-DL
4. Optional extensions
   - Fuzziness
   - Soft integrity constraints
     (expressing a kind of preferences)
   - ...

II. LP-style rules …

1. Scoped negations
   1.1. Well-founded semantics
        of negation as failure
   1.2. Answer-set (stable model) semantics
        - Can also be extended
          for "classical" negation

Note: We might just use stable models
and consider well-founded as
a more tractable approximation

II. … LP-style rules

2. Syntactic extensions
   2.1. Full Lloyd-Topor
   2.2. Frame notation with OIDs (cf. F-logic)

3. RDF and OWL-DL Interoperability
   - extended from Phase 1

4. Optional extensions
   - Disjunctive rules (answer set semantics)
   - Fuzziness
   - ...

III. Production rules

1. Allow retract etc. as rule actions
2. Allow procedural attachments and
   events in rule bodies
3. Also consider actions in the body
   a la Transaction Logic
   - has logical semantics unlike #III.1 and
     is compatible with LP-style semantics
4. Optional extensions
   - Fuzziness
   - ...