Linked Rules
Principles for Rule Reuse on the Web

Ankesh Khandelwal[1], Ian Jacobi[2], Lalana Kagal[2]

[1] **Tetherless World Constellation**
Rensselaer Polytechnic Institute

[2] **Decentralized Information Group**
Massachussets Institute of Technology

08/29/2011

Rule reuse

- Many sharable intelligent reasoning tasks
  - policies, regulations, business rules, work-flow plans, ontology mapping, email manipulation, foaf + rules ...
- Motivations for reuse
  - Reduce duplication of effort
  - Share common (semantics of) intelligent reasoning tasks
  - Reduce maintenance costs by minimizing the number of new rules defined for a particular reasoning task

Rule reuse is more accessible now

1. Ontology-based data modeling on the web
   - Ontology is shared conceptualization of domain providing shared vocabulary
   - Community development- leading to greater acceptance
2. Established methods for interoperability across rule systems
   - REWERSE Rule Markup Language (R2ML), RuleML, Rule Interchange Format (RIF)
   - Alleviate syntactic differences
   - Manifest common expressivities and features

Rule reuse is more accessible now

1. Ontology-based data modeling on the web
   - Ontology is shared conceptualization of domain providing shared vocabulary
   - Community development- leading to greater acceptance
2. Established methods for interoperability across rule systems
   - REWERSE Rule Markup Language (R2ML), RuleML, Rule Interchange Format (RIF)
   - Alleviate syntactic differences
   - Manifest common expressivities and features

Linked Data

The principles of Linked Data stated by Tim Berners-Lee :

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs, so that they can discover more things

Outline

Linked Rules

Principles for sharing rules on the Web such that they can be reused, combined, and extended on the Web in a manner similar to data (and ontologies) published in conformance with the *Linked Data* principles.

Linked Rules

Principles for sharing rules on the Web such that they can be reused, combined, and extended on the Web in a manner similar to data (and ontologies) published in conformance with the *Linked Data* principles.

1. Use HTTP URIs as names for rules
2. Standard Format for (Sharing) Rules: RIF
3. Standard Data Format: RDF
4. Link Rules

1. Name rules with HTTP URIs

Use HTTP URIs as names for rules, including individual rules, groups and rule-bases

- Refer to rule-bases for
  - use by reasoner
  - extension
  - associating metadata
- Refer to rules for
  - use in proof/justification
  - specialized reuse (constrained reuse)
  - priority specification

2. Standard format for (sharing) rules: RIF

When someone looks up (i.e. dereferences) rule definition, provide description in a standard format (RIF)

- RIF a W3C standard and completely webized language
- RIF specifies mechanism for compatibility with RDF
- RIF provides an infrastructure of datatypes and built-in functions
- RIF documents in RDF- rules as data

3. Standard data format: RDF

Define linked rules over data in standard data format (RDF)[1]

- Syntax and data format choices- *position*, *named argument* and *frame* terms- discourage re-usability
- Ontologies increasingly defined in RDF-based languages
- RDF $\rightarrow$ frame terms in RIF (blank-nodes $\sim$ local symbols); RDF $\rightarrow$ position terms in OWL 2 RL
- Some concerns such as representations of general n-ary relations and sequences, but next RDF is in pipeline
  - standardization of Turtle syntax, skolemization of blank nodes and graph identification

[1] **Linked rules**- rules published in conformance with Linked Rules principles

3. Standard data format: RDF

Define linked rules over data in standard data format (RDF)[1]

- Syntax and data format choices- *position*, *named argument* and *frame* terms- discourage re-usability
- Ontologies increasingly defined in RDF-based languages
- RDF $\rightarrow$ frame terms in RIF (blank-nodes $\sim$ local symbols); RDF $\rightarrow$ position terms in OWL 2 RL
- Some concerns such as representations of general n-ary relations and sequences, but next RDF is in pipeline
  - standardization of Turtle syntax, skolemization of blank nodes and graph identification

[1] **Linked rules**- rules published in conformance with Linked Rules principles

3. Standard data format: RDF

Define linked rules over data in standard data format (RDF)[1]

- Syntax and data format choices- *position*, *named argument* and *frame* terms- discourage re-usability
- Ontologies increasingly defined in RDF-based languages
- RDF $\rightarrow$ frame terms in RIF (blank-nodes $\sim$ local symbols); RDF $\rightarrow$ position terms in OWL 2 RL
- Some concerns such as representations of general n-ary relations and sequences, but next RDF is in pipeline
  - standardization of Turtle syntax, skolemization of blank nodes and graph identification

What about guarded rules?

$$r(x, y), r(y, z), \mathbf{s(x, y, z)} \rightarrow t(x, u), t(u, z)$$

[1] **Linked rules**- rules published in conformance with Linked Rules principles

4. Link rules

Link rules - so that they can be discovered, reused etc.

- Rules to rules
- Rules to ontologies
- Proofs/justifications to rules
- Rule descriptions (meta-data)

4. Link rules

Link rules - so that they can be discovered, reused etc.

- Rules to rules
- Rules to ontologies
- Proofs/justifications to rules
- Rule descriptions (meta-data)
  - **Terminologies used by rules**
  - **Dialect and complexity of language**
  - Expressive features used and restrictions obeyed
  - Ontologies for input vs output data
  - Purpose, creator
  - A VoID-like vocabulary

Outline

Ways of rule reuse

1. Simple reuse
2. Rule importation
3. Contextualized reuse
   - Scoped negation
   - Scoped fact matching/checking
   - Remote terms in RIF
   - *Generalized (dynamic) contextualized reuse*
4. Constrained reuse
   - Restrict (group of) one or more rules by restricting variables
   - Rules are modified by appending (more) conditions to rules
   - E.g. restrict `?x rdf:type :Student` by adding `?x rdf:type :GraduateStudent`

Ways of rule reuse

1. Simple reuse
2. Rule importation
3. Contextualized reuse
   - Scoped negation
   - Scoped fact matching/checking
   - Remote terms in RIF
   - *Generalized (dynamic) contextualized reuse*
4. Constrained reuse
   - Restrict (group of) one or more rules by restricting variables
   - Rules are modified by appending (more) conditions to rules
   - E.g. restrict `?x rdf:type :Student` by adding `?x rdf:type :GraduateStudent`

Ways of rule reuse

1. Simple reuse
2. Rule importation
3. Contextualized reuse
   - Scoped negation
   - Scoped fact matching/checking
   - Remote terms in RIF
   - *Generalized (dynamic) contextualized reuse*
4. Constrained reuse
   - Restrict (group of) one or more rules by restricting variables
   - Rules are modified by appending (more) conditions to rules
   - E.g. restrict `?x rdf:type :Student` by adding `?x rdf:type :GraduateStudent`

Support for rule reuse in RIF

- Remote term is not general enough (for contextualized reuse)
  - Restricted to single remote module
  - Modules must be known before-hand
- Lack of support for constrained reuse
  - Lack of support for distributed (individual) rule descriptions

Outline

Further work

- Mechanisms for establishing trust on rules
- Negotiating differences between diverse rule systems
- Vocabulary for rule descriptions (rule meta-data)
- Automated annotations of rule-bases, e.g., analyzing features used and not used in the rule-base
- Extensions to RIF for supporting features such as contextualized and constrained reuse of rules
- Implement AIR to RIF translation and other extensions, for Linked Rules-compliant publishing of rules using AIR

Being compliant with Linked Rules

How rule languages/systems (RLS) and authors can be compliant with Linked Rules

1. RLS support URI (or bnode) naming for every rule-base & rule
2. RLS have a corresponding RIF format, and support content-negotiation for retrieving descriptions in RIF and native language
3. RLS support RDF data format (directly or frame constructs etc.); Authors define rules over RDF data with an ontology model
4. RLS support links between rules. Rules described suitably for discovery and reuse
   - RLS support distributed rule-base (and rule) definitions
   - Rule(-base)s must be annotated with dialect and ontologies for target data of rules, by Authors or others. They may be annotated with other information such as purpose, author, features used or not used etc.

Acknowledgements

- We thank **Jim Hendler**, **Gregory Williams** and **Jiao Tao** for their feedback on this presentation
- We also thank **Harold Boley** for insightful comments on the paper

Linked Data and Linked Rules side-by-side

Linked Data	**Linked Rules**
Use URIs as names for things	Use HTTP URIs as names for rules- individual rules, groups and rule-bases
Use HTTP URIs so that people can look up those names	When someone looks up rule definition, provide description in a standard format (RIF)
When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)	Define linked rules over data in standard data format (RDF)
Include links to other URIs, so that they can discover more things	Link rules - so that they can be discovered, reused etc.

Ontologies in rules

- **Hybrid approach**: ontology terms used as constraints and checked against separate ontology language reasoner
- **Homogeneous approach**: unified language for ontology and rules. E.g. SWRL and ELP
  - For relatively less expressive ontology languages, like RDFS or OWL 2 RL, many rule systems can reason simultaneously with ontologies and rules

Interoperability via RuleML or RIF

- Rules are marked up in XML using schemas given by the interchange languages
- Rule systems define translations, often in XSLT, for translating rule-bases from and to the native representations of other rule systems

Basic constructs for data representation in RIF

- `family(?father, ?mother, ?child)` (position term)
- `family(:father→?father, :mother→?mother, :child→?child)` (named argument term)
- `?family#:Family[:father→?father, :mother→?mother, :child→?child]` (frame term)
- `?family rdf:type :Family; :father ?father; :mother ?mother; :child ?child .` (RDF in N3 syntax)

SPARQL-based condition matching

- Lot of information is available as Linked Data, e.g. hierarchical geographical relationships in Geonames dataset
- It is costly for in-memory reasoners to add millions of such RDF triples to local knowledge-base
- Alternatively, conditions may be checked online against SPARQL end-points (or local RDF stores)
- *Note*: This is not same as using SPARQL construct queries for rule descriptions

Reuse-able intelligent reasoning tasks

- Access control, accountability and data-usage policies
- Inter-organizational business rules, practices & regulations
- Medical decision support
- Work-flow plans
- Product compatibility
- Email manipulation, FOAF with person-centric rules, Reactive organizer
- Ontology mapping

Contextualized reuse

- Scoped negation

  `dataset not-contains fact?`

- Scoped fact matching/checking

  `dataset contains fact?`

- Remote terms in RIF

  `module entails fact?`

- Generalized (dynamic) contextualized reuse

  `(modules datasets) entail fact?`

Constrained reuse

- Restricting the scope of rule, by, e.g., adding explicit assumptions
- E.g. PRD-rule with variable declaration
  `Forall` $?v_1 \ldots ?v_n$ `such that` $(p_1 \ldots p_m)$ `(rule)`
  - $p_1 \ldots p_m$ can restrict scope for a predefined `rule`
  - E.g. `Forall ?customer such that (?customer # :ValuedCustomer) (:All-customers-rule)`