RuleML for Object-Relational Knowledge Representation on the Web

Harold Boley

Institute for Information Technology, National Research Council;
Faculty of Computer Science, University of New Brunswick, Canada

Developing F-logic and W3C RIF, PSOA RuleML permits relation applications with optional object identifiers and, orthogonally, positional or slotted arguments.

Introduction: Two IT Paradigms

Knowledge representation & problem solving in

- AI
- the (Semantic) Web
- IT at large

can be

1. Logic-based:
   FOL, Horn, LP
2. Object-oriented (and frame-based):
   CLOS, RDF, N3

Presentation Syntax: Rule Language

Example (PSOA RuleML business rule)

Adapts business rule from POSL logistics use case. Ternary `reciship` conclusion represents `reci`procal `ship`pings, at total cost (as single positional argument), between `source` and `dest`ination (as two slotted arguments). First two premises apply 4-ary `shipment` relation that uses anonymous cargo and named cost variables as two positional arguments, as well as `reciship`'s slotted arguments (in both 'directions').
Third premise is `External`-wrapped `numeric-add` RIF-DTB built-in applied on right-hand side of equality to sum up `shipment` costs for total. With the two facts, `?cost = ?57.0`.

```
Prefix(cpt  <http://eg.com/concepts#>)
Prefix(mus  <http://eg.com/museums#>)
Prefix(func <http://www.w3.org/2007/rif-builtin-function#>)
Prefix(xs   <http://www.w3.org/2001/XMLSchema#>)
Group (
  Forall ?cost ?cost1 ?cost2 ?A ?B (
    cpt:reciship(?cost cpt:source->?A cpt:dest->?B) :-
      And(cpt:shipment(? ?cost1 cpt:source->?A cpt:dest->?B)
          cpt:shipment(? ?cost2 cpt:source->?B cpt:dest->?A)
          ?cost = External(func:numeric-add(?cost1 ?cost2)) )
                                  )
  shipment("PC"^^xs:string "47.5"^^xs:float
           cpt:source->mus:BostonMoS cpt:dest->mus:LondonSciM)
  shipment("PDA"^^xs:string "9.5"^^xs:float
           cpt:source->mus:LondonSciM cpt:dest->mus:BostonMoS)
      )
```

Presentation Syntax: Rule Language (Cont'd)

Example (PSOA RuleML business rule, Cont'd)

The rule can be objectified as follows (`External`s are not being transformed):

```
Forall ?cost ?cost1 ?cost2 ?A ?B ?2 ?3 (
  Exists ?1 (?1#cpt:reciship(?cost cpt:source->?A cpt:dest->?B)) :-
    And(?2#cpt:shipment(? ?cost1 cpt:source->?A cpt:dest->?B)
        ?3#cpt:shipment(? ?cost2 cpt:source->?B cpt:dest->?A)
        ?cost = External(func:numeric-add(?cost1 ?cost2)) )
                                )
```

Further, it can be tupributed and slotributed (actually done by the semantics):

```
Forall ?cost ?cost1 ?cost2 ?A ?B ?2 ?3 (
  Exists ?1 (And(?1#cpt:reciship(?cost)
                 ?1#cpt:reciship(cpt:source->?A)
                 ?1#cpt:reciship(cpt:dest->?B))) :-
    And(?2#cpt:shipment(? ?cost1)
        ?2#cpt:shipment(cpt:source->?A)
        ?2#cpt:shipment(cpt:dest->?B)
        ?3#cpt:shipment(? ?cost2)
        ?3#cpt:shipment(cpt:source->?B)
        ?3#cpt:shipment(cpt:dest->?A)
        ?cost = External(func:numeric-add(?cost1 ?cost2)) )
                                )
```

Introduction: Psoa Terms and Rules

- Integration based on **p**ositional-**s**lotted, **o**bject-**a**pplicative (***psoa***) terms and rules
- Psoa term applies **function or predicate** symbol, possibly **instantiated by object**, to zero or more **positional or slotted (named)** arguments
- For a psoa term as **atomic formula**, predicate symbol is **class (type) of object** as well as **relation between arguments**, which describe object
- Each **argument** of a psoa term can be psoa term applying **function** symbol

Introduction: Psoa Rules Exemplified

Example (Rule-defined anonymous family frame)

`Group` is used to collect a rule and two facts. `Forall` quantifier declares orginal universal argument variables and generated universal OID variables `?2`, `?3`, `?4`. Infix `:-` separates conclusion from premises of rule, which derives anonymous/existential `family` frame from `married` relation `And` from `kid` relation of `husb` `Or` `wife` (the left-hand side is objectified on the right).

```
Group (                                  Group (
 Forall ?Hu ?Wi ?Ch (                     Forall ?Hu ?Wi ?Ch ?2 ?3 ?4 (
                                           Exists ?1 (
  family(husb->?Hu wife->?Wi child->?Ch):-  ?1#family(husb->?Hu wife->?Wi child->?Ch)) :-
   And(married(?Hu ?Wi)                      And(?2#married(?Hu ?Wi)
       Or(kid(?Hu ?Ch) kid(?Wi ?Ch))) )          Or(?3#kid(?Hu ?Ch) ?4#kid(?Wi ?Ch))) )
 married(Joe Sue)                         _1#married(Joe Sue)
 kid(Sue Pete)                            _2#kid(Sue Pete)
     )                                         )
```

Semantically, example is modeled by predicate extensions corresponding to following set of ground facts (the subdomain of individuals $\boldsymbol{D}_{\text{ind}}$ is to be defined):

{*o#family(husb->Joe wife->Sue child->Pete)* } $\cup$
{*_1#married(Joe Sue)*, *_2#kid(Sue Pete)*}, where $o \in \boldsymbol{D}_{\text{ind}}$.

Conclusion: Psoa Rules Made Horn

Example (Rule-extended named family frame)

Horn version of introductory example retrieves `family` frame with named OID variable in premise and uses its binding to extend that frame in conclusion (left: given; right: objectified).

```
Group (                                Group (
  Forall ?Hu ?Wi ?Ch ?o (                Forall ?Hu ?Wi ?Ch ?o ?1 ?2 (
    ?o#family(husb->?Hu                    ?o#family(husb->?Hu
              wife->?Wi                              wife->?Wi
              child->?Ch)      :-                    child->?Ch)      :-
      And(?o#family(husb->?Hu                And(?o#family(husb->?Hu
                    wife->?Wi)                             wife->?Wi)
          Or(kid(?Hu ?Ch)                        Or(?1#kid(?Hu ?Ch)
             kid(?Wi ?Ch))) )                       ?2#kid(?Wi ?Ch))) )
  inst4#family(husb->Joe                 inst4#family(husb->Joe
               wife->Sue)                             wife->Sue)
  kid(Sue Pete)                          _1#kid(Sue Pete)
      )                                      )
```

$\rightsquigarrow$ Simpler semantics corresponding to this set of ground facts:

{*inst4#family(husb->Joe wife->Sue child->Pete)*, *_1#kid(Sue Pete)*}