RuleML for Object-Relational Knowledge Representation on the Web

International Joint Conference on Artificial Intelligence, IJCAI 2011 Best Papers from Sister Conferences Track, Barcelona, Spain, July 19-22, 2011

Harold Boley

Institute for Information Technology, National Research Council; Faculty of Computer Science, University of New Brunswick, Canada
Introduction: Two IT Paradigms

Knowledge representation & problem solving in
- AI
- the (Semantic) Web
- IT at large

can be

1. **Relational (and logic-based):**
   - FOL, Horn, LP

2. **Object-oriented (and frame-based):**
   - CLOS, RDF, N3
Combined approaches:

- Description Logics (DLs)
- Object-Oriented Databases (OODBs) / Deductive Object-Oriented Databases (DOODs)
- Object-oriented logic languages: LIFE and Frame logic (F-logic)
- W3C Rule Interchange Format (RIF):
  - Semantics based on F-logic
  - Serialization syntax based on RuleML
F-logic and RIF extend first-order model-theoretic semantics for objects (frames)

Added separately from function and predicate applications to arguments

Resulting complexity of object-extended semantics can be reduced by integrating objects with applications
Integration based on **positional-slotted, object-applicative** rules of POSL and RuleML

F-logic’s model-theoretic semantics in the style of RIF is also the **starting point** of our **integrated semantics**

Permits **applications with optional object identifiers** and, orthogonally, **arguments** that are positional or slotted

Structured by these **independent dimensions** of defining features, language constructs can be freely combined
RuleML-2011 paper formalizes **positional-slotted, object-applicative (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
Psoa terms that apply a predicate symbol (as a relation) to *positional arguments* can be employed to make factual assertions.

An example, in simplified RIF (presentation) syntax, is term `married(Joe Sue)` for binary predicate `married` applied to *Joe* and *Sue*, where positional (left-to-right) order can be used to identify husband, as 1\(^{st}\) argument, and wife, as 2\(^{nd}\) argument.
Psoa terms that apply a predicate symbol (as a class) to *slotted arguments* correspond to typed attribute-value descriptions.

An example is psoa term

```
family(husb->Joe wife->Sue) or
family(wife->Sue husb->Joe) for
family-typed attribute-value pairs (slots)
{<husb,Joe>, <wife,Sue>}
```

Easily extended with further slots, e.g. by adding children, as in 
```
family(husb->Joe wife->Sue child->Pete)
```
Usually, slotted terms describe an object symbol, i.e. an object identifier (OID), maintaining object identity even when slots of their descriptions are added or deleted.

This leads to (typed) frames in the sense of F-logic.

E.g., using RIF’s membership syntax #, OID inst1 in class family is describable by inst1#family(husb->Joe wife->Sue), inst1#family(husb->Joe wife->Sue child->Pete), etc. Psoa terms can also specialize to class membership terms, e.g. inst1#family(), abridged inst1#family, represents inst1 ∈ family.
Like OID-describing slotted terms constitute a (multi-slot) ‘frame’, positional terms that describe an object constitute a (single-tuple) ‘shelf’, similar to a (one-dimensional) array describing its name.

Thus, family’s husb and wife slots can be positionalized as in earlier married example: \texttt{inst1#family(Joe Sue)} describes \texttt{inst1} with tuple [Joe Sue]
Combined positional-slotted psoa terms are allowed, similarly as in XML elements (tuple \(\rightarrow\) subelements, slots \(\rightarrow\) attributes), e.g. describing an object, as in RDF descriptions (object \(\rightarrow\) subject, slots \(\rightarrow\) properties).

For example, \texttt{inst1\#family(Joe Sue child->Pete)} describes \texttt{inst1} with two positional and one slotted argument.
An atomic formula without OID is treated as having implicit OID.

An OID-less application is objectified by syntactic transformation: The OID of a ground fact is new constant generated by ‘new local constant’ (stand-alone _); the OID of non-ground fact or atomic formula in rule conclusion, \( f(\ldots) \), is new, existentially scoped variable \(?i\); leading to \( \text{Exists} \ ?i \ (?i\#f(\ldots)) \); the OID of other atomic formulas is new variable generated by ‘anonymous variable’ (stand-alone ?).

Objectification allows compatible semantics for an atom constructed as RIF-like slotted (named-argument) term and corresponding frame, solving issue with named-argument terms:

http://lists.w3.org/Archives/Public/public-rif-wg/2008Jul/0000.html
For example, slotted-fact assertion
family(husb->Joe wife->Sue) is
syntactically objectified to assertion
_#family(husb->Joe wife->Sue), and
— if _1 is first new constant from _1, _2, ... — to
_1#family(husb->Joe wife->Sue)

This typed frame, then, is semantically
*slotributed* to _1#family(husb->Joe) and
_1#family(wife->Sue)
Rules can be defined on top of psoa terms in a natural manner.

A rule derives (a conjunction of possibly existentially scoped) conclusion psoa atoms from (a formula of) premise psoa atoms.

Consider example with rule deriving family frames.
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**

\[
\text{Forall } ?Hu \text{ ?Wi ?Ch } (\text{family(husb->}) ?Hu \text{ wife->}) ?Wi \text{ child->}) ?Ch) :- \text{And}(\text{married(?Hu ?Wi)} \text{ Or}(\text{kid(?Hu ?Ch)} \text{ kid(?Wi ?Ch)}))
\]

\[
\text{married(Joe Sue)}
\text{kid(Sue Pete)}
\]

**Group**

\[
\text{Forall } ?Hu \text{ ?Wi ?Ch ?2 ?3 ?4 } (\text{Exists } ?1 (\text{family(husb->}) ?Hu \text{ wife->}) ?Wi \text{ child->}) ?Ch) :- \text{And}(\text{married(?Hu ?Wi)} \text{ Or}(\text{kid(?Hu ?Ch)} \text{ kid(?Wi ?Ch)}))
\]

\[
_1\text{married(Joe Sue)}
_2\text{kid(Sue Pete)}
\]

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

\[
\{ o\text{#family(husb->Joe wife->Sue child->Pete)} \} \cup
\{ _1\text{married(Joe Sue)}, _2\text{kid(Sue Pete)} \}, \text{ where } o \in D_{\text{ind}}.
\]
PSOA RuleML is defined here as a language incorporating this integration:

- PSOA RuleML’s human-readable presentation syntax
- PSOA RuleML’s model-theoretic semantics
- Conclusion and future work
In this definition, \textit{base term} means a simple term, an anonymous psoa term (i.e., an anonymous frame term, single-tuple psoa term, or multi-tuple psoa term), or a term of the form \texttt{External(t)}, where \( t \) is an anonymous psoa term. Anonymous term can be \textit{deobjectified} (by omitting main \(?#\)) if its re-objectification results in old term (i.e., re-introduces \(?#\)).

**Definition (Term)**

1. \textit{Constants and variables}. If \( t \in \text{Const} \) or \( t \in \text{Var} \) then \( t \) is a \textit{simple term}

2. \textit{Equality terms}. \( t = s \) is an \textit{equality term} if \( t, s \) are base terms

3. \textit{Subclass terms}. \( t##s \) is a \textit{subclass term} if \( t, s \) are base terms

4. \textit{Positional-slotted, object-applicative terms.} \[
o#f\left([t_1,1...t_1,n_1]...[t_m,1...t_m,n_m]\ p_1->v_1...p_k->v_k\right)
\]
   is a \textit{positional-slotted, object-applicative (psoa) term} if 
   \( f \in \text{Const} \) and \( o, t_1, 1, ..., t_1, n_1, ..., t_m, 1, ..., t_m, n_m, 
   p_1, ..., p_k, v_1, ..., v_k, m \geq 0, k \geq 0 \), are base terms
For \( m = 1 \) psoa terms become \textbf{single-tuple psoa terms}
\[ o\#f([t_{1,1} \ldots t_{1,n_1}] p_1\rightarrow v_1 \ldots p_k\rightarrow v_k), \] abridged to
\[ o\#f(t_{1,1} \ldots t_{1,n_1} p_1\rightarrow v_1 \ldots p_k\rightarrow v_k) \]
These can be further specialized in two ways, which can be orthogonally combined:

- For \( o \) being the anonymous variable \(?\), they become \textbf{anonymous single-tuple psoa terms}
\[ ?\#f(t_{1,1} \ldots t_{1,n_1} p_1\rightarrow v_1 \ldots p_k\rightarrow v_k), \] deobjectified \( f(t_{1,1} \ldots t_{1,n_1} p_1\rightarrow v_1 \ldots p_k\rightarrow v_k) \). These can be further specialized:
  - For \( k = 0 \), they become \textbf{positional terms}
\[ ?\#f(t_{1,1} \ldots t_{1,n_1}), \] deobjectified \( f(t_{1,1} \ldots t_{1,n_1}) \), corresponding to the usual terms and atomic formulas of classical first-order logic
  - For \( f \) being the root class \( \text{Top} \), they become \textbf{untyped single-tuple psoa terms}
\[ o\#\text{Top}(t_{1,1} \ldots t_{1,n_1} p_1\rightarrow v_1 \ldots p_k\rightarrow v_k). \] These can be further specialized:
    - For \( k = 0 \), they become \textbf{untyped single-tuple shelf terms}
\[ o\#\text{Top}(t_{1,1} \ldots t_{1,n_1}) \] describing object \( o \) with positional arguments \( t_{1,1}, \ldots, t_{1,n_1} \)
Definition (Formula, Rule Language)

3 Rule implication: $\varphi : \neg \psi$ is a formula, called \textit{rule implication}, if:

- $\varphi$ is a head formula or a \textit{conjunction} of head formulas, where a head formula is an atomic formula or an \textit{existentially} scoped atomic formula,
- $\psi$ is a condition formula, and
- none of the atomic formulas in $\varphi$ is an externally defined term (i.e., term of the form $\text{External}(\ldots)$)

4 Universal rule: If $\varphi$ is a rule implication and $?V_1, \ldots, ?V_n$, \(n>0\), distinct variables then $\text{Forall } ?V_1 \ldots ?V_n(\varphi)$ is a \textit{universal rule} formula. It is required that all free variables in $\varphi$ occur among variables $?V_1 \ldots ?V_n$ in quantification part. Generally, an occurrence of variable $?v$ is \textit{free} in $\varphi$ if it is not inside subformula of $\varphi$ of the form $\exists ?v(\psi)$ and $\psi$ is a formula. Universal rules are also referred to as \textit{PSOA RuleML rules}. 
Use \( TV \) as set of semantic truth values \( \{t, f\} \)

Truth valuation of PSOA RuleML formulas will be defined as mapping \( TVal_I \) in two steps:

1. Mapping \( I \) generically bundles various mappings from semantic structure, \( I \);
   - \( I \) maps formula to element of domain \( D \)

2. Mapping \( I_{\text{truth}} \) takes such a domain element to \( TV \)

This indirectness allows HiLog-like generality
A semantic structure, $\mathcal{I}$, is a tuple of the form $< TV, DTS, D, D_{ind}, D_{func}, I_C, I_V, I_{psoa}, I_{sub}, I_\mathbf{=}, I_{external}, I_{truth}>$

Here $D$ is a non-empty set of elements called the domain of $\mathcal{I}$, and $D_{ind}, D_{func}$ are nonempty subsets of $D$

The domain must contain at least the root class: $\top \in D$

$D_{ind}$ is used to interpret elements of Const acting as individuals

$D_{func}$ is used to interpret constants acting as function symbols

As before, Const denotes set of all constant symbols and Var set of all variable symbols

DTS denotes set of identifiers for primitive datatypes
Definition (Semantic structure, Cont’d)

3 \( I_{\text{psoa}} \) maps \( D \) to total functions \( D_{\text{ind}} \times \text{SetOfFiniteBags}(D^*_{\text{ind}}) \times \text{SetOfFiniteBags}(D_{\text{ind}} \times D_{\text{ind}}) \rightarrow D \). Interprets psoa terms, combining positional, slotted, and frame terms, as well as class memberships. Argument \( d \in D \) of \( I_{\text{psoa}} \) represents function or predicate symbol of positional terms and slotted terms, and object class of frame terms, as well as class of memberships. Element \( o \in D_{\text{ind}} \) represents object of class \( d \), which is described with two bags.

- Finite bag of finite tuples \( \{<t_{1,1}, ..., t_{1,n_1}>, ..., <t_{m,1}, ..., t_{m,n_m}>\} \in \text{SetOfFiniteBags}(D^*_{\text{ind}}) \), possibly empty, represents positional information. \( D^*_{\text{ind}} \) is set of all finite tuples over the domain \( D_{\text{ind}} \). Bags are used since order of tuples in a psoa term is immaterial and tuples may repeat.

- Finite bag of attribute-value pairs \( \{<a_1, v_1>, ..., <a_k, v_k>\} \in \text{SetOfFiniteBags}(D_{\text{ind}} \times D_{\text{ind}}) \), possibly empty, represents slotted information. Bags, since order of attribute-value pairs in a psoa term is immaterial and pairs may repeat.
Definition (Semantic structure, Cont’d)

Generic mapping from terms to $D$, denoted by $I$

- $I(k) = I_C(k)$, if $k$ is a symbol in $\text{Const}$
- $I(?v) = I_V(?v)$, if $?v$ is a variable in $\text{Var}$
- $I(\circ \# f([t_{1,1} \ldots t_{1,n_1}] \ldots [t_{m,1} \ldots t_{m,n_m}] \ a_1 \rightarrow v_1 \ldots a_k \rightarrow v_k)) = I_{psoa}(I(f))(I(\circ), \{<I(t_{1,1}), \ldots, I(t_{1,n_1} )>, \ldots, <I(t_{m,1}), \ldots, I(t_{m,n_m} )>\},$
  \{<I(a_1), I(v_1)>, \ldots, <I(a_k), I(v_k)>)\}$

Again {...} denote bags of tuples and attribute-value pairs.

- $I(c_1 \# \# c_2) = I_{sub}(I(c_1), I(c_2))$
- $I(x=y) = I_{=}(I(x), I(y))$
- $I(\text{External}(p(s_1 \ldots s_n))) = I_{external}(p)(I(s_1), \ldots, I(s_n))$
Define mapping, $TVal_{\mathcal{I}}$, from set of all non-document formulas to $TV$

For atomic formula $\phi$, $TVal_{\mathcal{I}}(\phi)$ defined essentially as $I_{\text{truth}}(I(\phi))$

Recall that $I(\phi)$ is just an element of domain $D$ and $I_{\text{truth}}$ maps $D$ to truth values in $TV$

HiLog-style definition inherited from RIF-FLD and equivalent to a standard one for first-order languages such as RIF-BLD and PSOA RuleML — but enables future higher-order features
**Semantics: Interpretation of Formulas**

**Definition (Truth valuation)**

*Truth valuation* for well-formed formulas in PSOA RuleML determined using function $TVal_I$:

**Psoa formula:**

$$TVal_I(o#f([t_1,1...t_1,n_1]...[t_m,1...t_m,n_m] a_1->v_1...a_k->v_k)) = I_{truth}(I(o#f([t_1,1...t_1,n_1]...[t_m,1...t_m,n_m] a_1->v_1...a_k->v_k)))$$

The formula consists of an object-typing membership, a bag of tuples representing a conjunction of all the object-centered tuples (*tupribution*), and a bag of slots representing a conjunction of all the object-centered slots (*slotribution*). Hence use restriction, where $m \geq 0$ and $k \geq 0$:

- $TVal_I(o#f([t_1,1...t_1,n_1]...[t_m,1...t_m,n_m] a_1->v_1...a_k->v_k)) = t$ if and only if
  - $TVal_I(o#f) = TVal_I(o#Top([t_1,1...t_1,n_1])) = ... = TVal_I(o#Top([t_m,1...t_m,n_m])) = TVal_I(o#Top(a_1->v_1)) = ... = TVal_I(o#Top(a_k->v_k)) = t$
Definition (Truth valuation, Cont’d)

8 Rule implication:

- $\text{TVal}_I(\text{conclusion} : - \text{condition}) = t$, if either
  - $\text{TVal}_I(\text{conclusion}) = t$ or $\text{TVal}_I(\text{condition}) = f$
  - $\text{TVal}_I(\text{conclusion} : - \text{condition}) = f$ otherwise

9 Groups of rules:

If $\Gamma$ is a group formula of the form $\text{Group}(\varphi_1 \ldots \varphi_n)$ then

- $\text{TVal}_I(\Gamma) = t$ if and only if $\text{TVal}_I(\varphi_1) = \ldots = \text{TVal}_I(\varphi_n) = t$
- $\text{TVal}_I(\Gamma) = f$ otherwise

In other words, rule groups are treated as conjunctions.
W3C’s RIF-BLD has provided a reference semantics for extensions, and for continued efforts, as described here.

Project with Alexandre Riazanov is implementing PSOA RuleML in Vampire Prime via TPTP.
Conclusion: Horn

Further efforts concern Horn rules

Notice introductory example is not Horn in that there is a head existential after objectification

To address this issue, it can be modified as follows
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).

\[
\text{Group (}
\begin{align*}
\forall ?Hu \ ?Wi \ ?Ch \ ?o & : \\
?o\#\text{family}(\text{husb->}?Hu & \\
\text{wife->}?Wi & \\
\text{child->}?Ch) & : - \\
\land(?o\#\text{family}(\text{husb->}?Hu & \\
\text{wife->}?Wi) & \\
\lor(\text{kid}(?Hu \ ?Ch) & \\
\text{kid}(?Wi \ ?Ch)) & ) \\
\text{inst4}\#\text{family}(\text{husb->}Joe & \\
\text{wife->}Sue) & \\
\text{kid}(Sue \ Pete) & )
\end{align*}
\]

\[
\text{Group (}
\begin{align*}
\forall ?Hu \ ?Wi \ ?Ch \ ?o \ ?1 \ ?2 & : \\
?o\#\text{family}(\text{husb->}?Hu & \\
\text{wife->}?Wi & \\
\text{child->}?Ch) & : - \\
\land(?o\#\text{family}(\text{husb->}?Hu & \\
\text{wife->}?Wi) & \\
\lor(?1\#\text{kid}(?Hu \ ?Ch) & \\
?2\#\text{kid}(?Wi \ ?Ch)) & ) \\
\text{inst4}\#\text{family}(\text{husb->}Joe & \\
\text{wife->}Sue) & \\
_1\#\text{kid}(Sue \ Pete) & )
\end{align*}
\]

\[\Rightarrow\] Simpler semantics corresponding to this set of ground facts:

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