The Integrated PSOA RuleML for Interoperating SQL Relations and SPARQL Graphs

Harold Boley, Faculty of Computer Science, University of New Brunswick, Canada

The 6th Atlantic Workshop on Semantics and Services (AWoSS 2015)
Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, December 9, 2015

Data is often stored in one of two paradigms: As relations (predicate-centered), e.g. in the SQL-queried Deep Web, or as graphs (object-centered), e.g. in the SPARQL-queried Semantic Web. This divide has also led to separate relational and object-centered rule paradigms for processing the data. Projects involving both relations and graphs have thus been impeded by the paradigm boundaries, from modeling to implementation. These boundaries can be dissolved with the integrated language Positional-Slotted, Object-Applicative (PSOA) RuleML. PSOA RuleML permits a relation application to have an Object IDentifier (OID) – typed by the relation as its class – and, orthogonally, the relation's arguments to be positional or slotted. The resulting novel concept of a positional-slotted, object-applicative (psoa) atomic formula can be employed as follows:

1. Relationship (Prolog-like relation application): Predicate-centered, positional atom without an OID and with an – ordered – sequence of arguments
2. Shelf (Object-IDentified relationship): Object-centered, positional atom with an OID and with a sequence of arguments
3. Pairship (RIF-like named-argument term): Predicate-centered, slotted atom without an OID and with an – unordered – multi-set of slots (each being a pair of a name and a filler)
4. Frame (F-logic-like typed graph-node OID and its outgoing-arrow slots): Object-centered, slotted atom with an OID and with a multi-set of slots

Via atoms, rules of the form {relationship|shelf|pairship|frame} if {relationship|shelf|pairship|frame}* can be built, where a conclusion, consisting of any of the four kinds of psoa atoms, is derived from a condition of conjuncts (here indicated by a "*") of arbitrary psoa atoms. Exemplifying the advanced form of (conclusion-)existential rules, frames (whose existential OIDs are generated on-the-fly for each rule invocation) can be derived as needed from a conjunction of relationships (cf. the Family Example).

PSOA RuleML uses visualization (Grailog), presentation, and serialization syntaxes. Its model-theoretic semantics blends (OID-over-)slot distribution and integrated psoa terms. Efficient implementations, instantiating the PSOATransRun framework, translate PSOA RuleML knowledge bases and queries to TPTP (PSOA2TPTP) or Prolog (PSOA2Prolog). A tutorial-style overview gives details of the language.

A use case on bidirectional SQL-PSOA-SPARQL transformation for interoperability will be presented. Its core transformation between the relational paradigm (SQL) and the object-centered paradigm (SPARQL) is expressed in a language-internal manner within PSOA RuleML itself.