PSOA RuleML
Bridges Graph and Relational Databases

Harold Boley
Faculty of Computer Science, University of New Brunswick
Fredericton, Canada
harold[DT]boley[AT]unb[DT]ca

Abstract

In PSOA RuleML, Graph Databases and Relational Databases are bridged conceptually, with interoperation paths through its metamodel of three orthogonal dimensions, as well as programmatically, with transition rules realized in PSOATransRun.

1 Introduction

The gap between the graph/object vs. relational paradigms has been a major concern in data modeling, as evidenced, e.g., by object-relational mapping, R2RML, NoSQL, and Multi-Model Databases. PSOA RuleML [2] bridges the paradigms of Graph Databases [1] and Relational Databases [5], augmented by paradigm-integrating knowledge (rule) bases on top of the paradigm-integrating data (facts): PSOA RuleML's databases (fact bases) generalize the instance level of Graph and Relational Databases; its knowledge bases complement facts by rules for deductive retrieval (extending the Datalog-level, function-free expressiveness of Deductive Databases to the Horn-logic expressiveness of Logic Programming), interoperation, and general computation, as well as for optionally emulating part of the schema level.

The current document discusses overarching graph-relational topics. This is in continuation of earlier work on graph/object-relational interoperation with PSOA RuleML [3].

PSOA RuleML's two bridge-building starting points are described in two dedicated documents, which can be read in either order:

- PSOA RuleML Meets Graph Databases
- PSOA RuleML Meets Relational Databases

In these documents, the paradigmatic notions of graph and relation are both differentiated:

- Graphs can be a) directed labeled graphs or b) generalized graphs (including directed labelnode hypergraphs)
- Relations can have tables with a) the original tuple-like rows or b) the established record-like rows

The b) parts of both paradigms indicate a graph-relational convergence, which prepares PSOA RuleML's bridge building.

2 Metamodel Paths

The PSOA RuleML metamodel in [4], Appendix A, Fig. 5, permits 'shortest paradigm-bridging paths' through its 3-dimensional qualitative design space of eighteen unit cubes (units). Such a path is a composition of (unit-to-adjacent-unit) transitions, where each transition changes the value in a dimension (to some of the one or two other possible values). Since these transitions constitute elementary steps, they make all intermediate units explicit as the most fine-grained 'conceptual pillars' for bridging any two units. We highlight an exemplary shortest pv1-pn4 path between the purest data/knowledge forms of the paradigms, which starts with the table rows of Relational Databases and, even within this paradigm, does the initial pv1-pv3 transition from the unit for the original tuple-like rows ("relationships") to the unit for the established record-like rows ("pairships"):

Metamodel path (four units with three transitions):

1. (pv1-constraining) oidless, single-tuple, perspectival atoms, called "relationships"
   - pv1-pv3 transition: single-tuple → slotted
2. (pv3-corresponding) oidless, slotted, perspectival atoms, called "pairships"
   - pv3-pv4 transition: oidless → oidful
3. (pv4-realizing) oidful, slotted, perspectival atoms
   - pv4-pn4 transition: perspectival → perspeneutral
4. (pn4-corresponding) oidful, slotted, perspeneutral atoms, called "frames"

The path ends at the unit for the typed-node-with-independent-labeled-outgoing-arc atoms ("frames") of Graph Databases, reached by the final pv4-pn4 transition. The (unique) inverse pn4-pv1 path is obtained by inverted transitions (leading from unit 4. to 1.).

These metamodel notions can be illustrated with simple sample atoms about a traditional wedding, where different predicates Wedding1, ..., Wedding4 are used for clarity here (for the visualization of the eighteen kinds of atoms including these four see PSOAMetamodelGrailogWedding.pdf):

Example path (WeddingI atoms with abstract transitions):

1. Wedding1(Mary John) *or* Wedding1(+[Mary John])
   - pv1-pv3 transition: single-tuple → slotted
2. Wedding2(bride+>Mary groom+>John)
   - pv3-pv4 transition: oidless → oidful
3. w31#Wedding3(bride+>Mary groom+>John)
   - pv4-pn4 transition: perspectival → perspeneutral
4. w41#Wedding4(bride->Mary groom->John)

The Object IDentifiers (OIDs) w31 and w41 can be seen as the Skolem constants resulting from explicit existential wrappers of corresponding atoms Exists ?k (?k#Wedding3(bride+>Mary groom+>John)) and Exists ?k (?k#Wedding4(bride->Mary groom->John)), respectively. The pv3-pv4 transition exemplifies nine oidless → oidful transitions that map the subcube for the oidless half of the metamodel cube to its oidful subcube such that each oidless *atom* is equivalent to Exists ?k (?k#*atom*), where the variable ?k does not occur in *atom*.

While the slot groom+>John is dependent on the Wedding2 and Wedding3 predicates of its perspectival atoms, the slot groom->John is independent from the Wedding4 predicate of its perspeneutral atom. Thus, if the wedding was celebrated at a derby, an additional atom w31#Derby(groom+>James) would create a multi-membership for w31 such that w31 as a Derby has groom James but w31 as a Wedding3 has groom John; however, an additional atom w41#Derby(groom->James) would create a multi-membership for w41 such that w41 has a multi-valued groom slot not distinguishing James and John for Derby and Wedding4, respectively (for a PSOATransRun-executable database and queries see MetamodelWeddingDerby.psoa).

3 Transition Rules

PSOA RuleML rules can be defined for the transitions between the sample atoms, where (for 1 ≤ I ≤ 3) an atom with predicate WeddingI is used as a fact for a rule with a unifying condition, and the atom with predicate WeddingI+1 is used as a query derived by the rule conclusion:

Example path (WeddingI atoms with implicit-Forall rules):

1. Wedding1(Mary John) *or* Wedding1(+[Mary John])
   - Wedding2(bride+>?x groom+>?y) :- Wedding1(?x ?y)
2. Wedding2(bride+>Mary groom+>John)
   - f3(?x ?y)#Wedding3(bride+>?x groom+>?y) :-
     Wedding2(bride+>?x groom+>?y), where w31 = f3(Mary John)
3. w31#Wedding3(bride+>Mary groom+>John)
   - f4(?o)#Wedding4(bride->?x groom->?y) :-
     ?o#Wedding3(bride+>?x groom+>?y), where w41 = f4(w31)
4. w41#Wedding4(bride->Mary groom->John)

The composition of these rules can be abridged to a single PSOA rule defined for the overall pv1-pn4-transition path between the sample atoms (the explicit Forall allows copy & paste for execution by PSOATransRun, developed in [6]):

```
% Relationship-to-Frame Rule (R2F):
Forall ?x ?y (
  f5(?x ?y)#Wedding4(bride->?x groom->?y) :- Wedding1(?x ?y)
)
% Here w41 = f5(Mary John)
```

Likewise, the composition for the overall inverse pn4-pv1-transition path can be abridged (the stand-alone "?", as PSOA's anonymous variable, is used to discard the frame condition's OID, not needed in the relationship conclusion):

```
% Frame-to-Relationship Rule (F2R):
Forall ?x ?y (
  Wedding1(?x ?y) :- ?#Wedding4(bride->?x groom->?y)
)
```

Only the rules R2F and F2R are required for graph-relational WeddingI-atom interoperation (PSOATransRun assumes local constants, hence "_"-prefixes them in answer bindings): Based on a fact Wedding1(Mary John), rule R2F succeeds for queries f5(Mary John)#Wedding4(bride->Mary groom->John) and ?o#Wedding4(bride->Mary groom->?b), the latter with bindings ?o=_f5(_Mary _John), ?b=_John; based on a fact f5(Mary John)#Wedding4(bride->Mary groom->John), rule F2R succeeds for queries Wedding1(Mary John) and Wedding1(?a ?b), the latter with bindings ?a=_Mary, ?b=_John (for a PSOATransRun-executable knowledge base and queries see MetamodelWeddingInterop.psoa). These rules thus permit round-tripping that keeps atoms unchanged for a given OID-name ground instantiation of f5(?x ?y) such as, with our sample facts, for f5(Mary John).

4 Conclusions

PSOA RuleML constitutes a paradigm-integrating language permitting expressive data and knowledge representation, translation, and execution (all with PSOATransRun) across the major graph-relational paradigms for databases as well as knowledge bases. The integration of different paradigms in one language facilitates the interoperation of content between these paradigms. The perspectivity-sliced visualization, in [4], of the graph-relational-bridging metamodel cube is currently being complemented by user-selected (JavaScript-implemented) descriptor- and OID-sliced visualizations in a collaboration between RuleML Inc. and the UNB FCS HCI Lab. An entry point into the open development of PSOA RuleML and PSOATransRun is the PSOATransRun Development Agenda.

References

[1] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3Logic: A Logical Framework for the World Wide Web. *Theory and Practice of Logic Programming (TPLP)*, 8(3), May 2008. URL: https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/n3logic-a-logical-framework-for-the-world-wide-web/5CB102B7E35457C8D07EC2B8281C8317.

[2] Harold Boley. A RIF-Style Semantics for RuleML-Integrated Positional-Slotted, Object-Applicative Rules. In *Proc. 5th International Symposium on Rules: Research Based and Industry Focused (RuleML-2011 Europe), Barcelona, Spain*, Lecture Notes in Computer Science, pages 194–211. Springer, July 2011. URL: https://link.springer.com/chapter/10.1007/978-3-642-22546-8_16.

[3] Harold Boley. PSOA RuleML: Integrated Object-Relational Data and Rules. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules (RuleML 2015) - 11th International Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*. Springer, 2015. URL: https://link.springer.com/chapter/10.1007/978-3-319-21768-0_5.

[4] Harold Boley and Gen Zou. Perspectival Knowledge in PSOA RuleML: Representation, Model Theory, and Translation. *CoRR*, abs/1712.02869, 2017. URL: http://arxiv.org/abs/1712.02869, `arXiv:1712.02869`.

[5] David Maier. *The Theory of Relational Databases*. CS Press, 1983. URL: http://web.cecs.pdx.edu/~maier/TheoryBook/TRD.html.

[6] Gen Zou. *Translators for Interoperating and Porting Object-Relational Knowledge*. PhD thesis, Faculty of Computer Science, University of New Brunswick, April 2018. URL: https://unbscholar.lib.unb.ca/islandora/object/unbscholar%3A9279.