Data Systematics: The PSOA RuleML Metamodel Illustrated by Grailog Visualization of Wedding Atoms

(PDF version: ruleml.org/talks/PSOAMetamodelGrailogWedding.pdf**)**

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

July 13, 2018
Update: Oct. 16, 2019

Introduction

- PSOA RuleML builds on a novel data systematics
- Slicing and dicing the *PSOA metamodel cube* (from PSOAPerspectivalKnowledge, Appendix A)
- Exemplify with oidless/oidful, tupled/slotted/combined, independent/dependent/combined atoms (2*3*3 = 18)
- Illustrate all kinds of atoms by Grailog visualization, realizing them in *presentation syntax* by PSOATransRun
- Informal syntax templates and English semantics (formal in PSOAPerspectivalKnowledge, Sections 4 and 5)
- Experience full metamodel dynamically by online PSOAMetaViz visualization, realized in JavaScript/JSON

Slicing and Dicing the PSOA Metamodel Cube

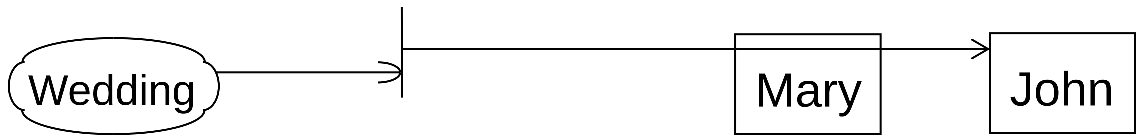
- Via 3 (orthogonal) dimensions, the **full metamodel** cube systematizes 18 kinds of atoms that are contained in 18 unit cubes (units) named in*j*, de*j*, id*j* (*j*=1,...,6)
- Choosing one of the reductions DVO, VDO, or OVD (s. below), users can slice and dice the cube, in a kind of (meta)OLAP, initially reducing its 3 dimensions to slices of 2 dimensions:
- DVO reduction, via Dependency dimension, to 3 slices, each with 6 units structured by Variety-row and OID-column dimensions:
  - 6 **in**dependent units **in***j* (*j*=1,...,6) vs. 6 **de**pendent units **de***j* (*j*=1,...,6) vs. 6 combined **i**ndependent+**d**ependent units **id***j* (*j*=1,...,6)
- The **core metamodel** is an 8-unit subcube of the full metamodel cube, which can be reduced, DVO-style, to 2 Dependency slices: in1-in4 and de1-de4
  - Each includes a 'landmark' unit: ***framepoint*** atoms (in4) and ***relationship*** atoms (de1)
- VDO reduction (e.g., for full metamodel), via Variety dimension, to 3 slices, each with 6 units structured by Dependency-row and OID-column dimensions:
  - 6 tupled+slotted units in*j*, de*j*, id*j* (*j*=5,6) vs. 6 slotted units in*j*, de*j*, id*j* (*j*=3,4) vs. 6 tupled units in*j*, de*j*, id*j* (*j*=1,2)
- OVD reduction (e.g., for full metamodel), via OID dimension, to 2 slices, each with 9 units structured by Variety-row and Dependency-column dimensions:
  - 9 oidful units in*j*, de*j*, id*j* (*j*=2,4,6) vs. 9 oidless units in*j*, de*j*, id*j* (*j*=1,3,5)

Exemplifying the Dependency Slices

Core oidless/oidful, tupled/slotted atoms that are **in**dependent:

Grailog:

in1. single-tuple: shelfships

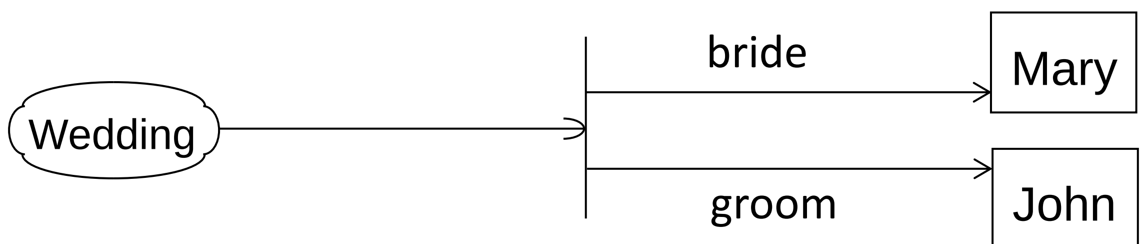


Wedding(-[Mary John])

in2. single-tuple: shelfpoints

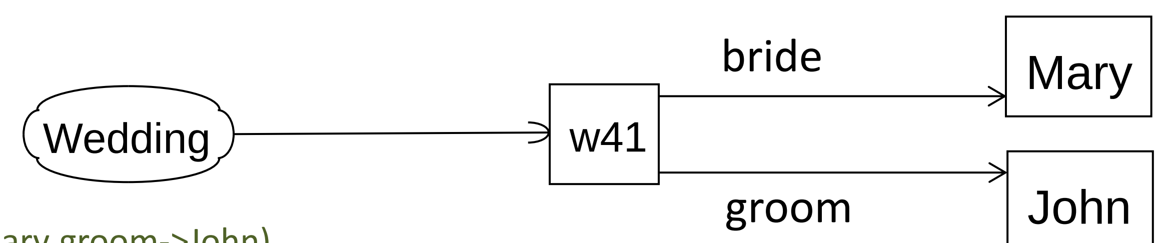
w21#Wedding(-[Mary John])

in3. frameships



Wedding(bride->Mary groom->John)

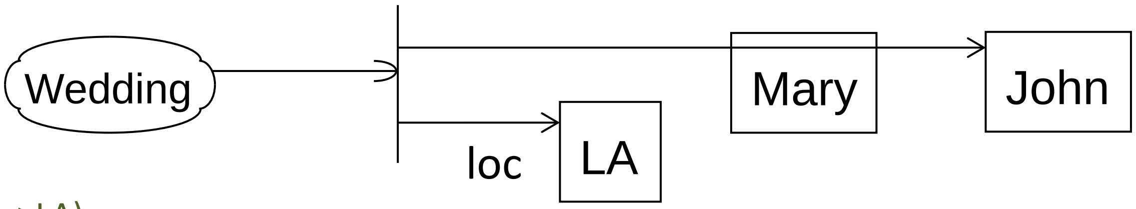
in4: **framepoints**



w41#Wedding(bride->Mary groom->John)

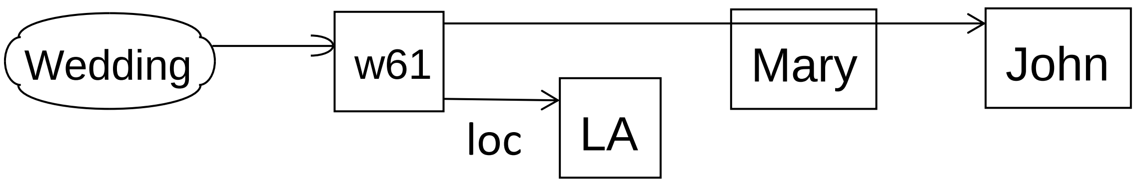
Extra oidless/oidful, combined tupled+slotted atoms that are **in**dependent:

in5. single-tuple:
shelframeships



Wedding(-[Mary John] loc->LA)

in6. single-tuple:
shelframepoints



w61#Wedding(-[Mary John] loc->LA)

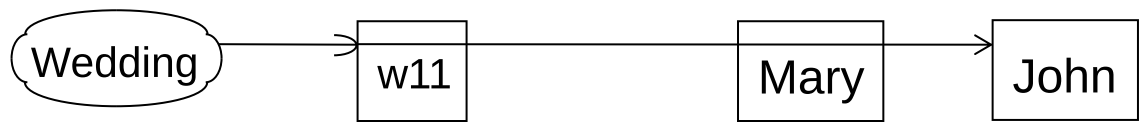
Core oidless/oidful, tupled/slotted atoms that are **de**pendent:

de1. single-tuple:
relationships

or

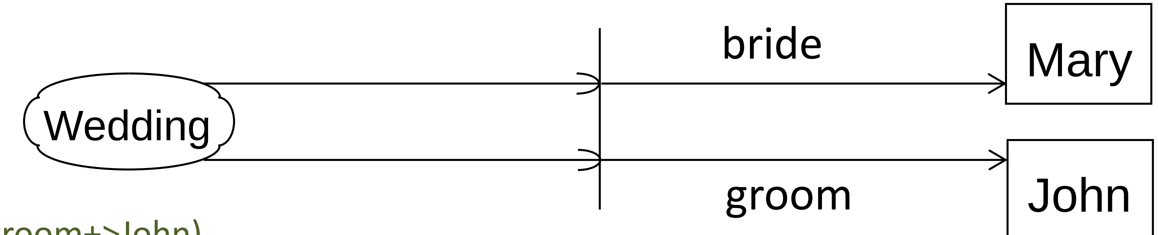
Wedding(Mary John) *or* Wedding(+[Mary John])

de2. single-tuple:
relationpoints



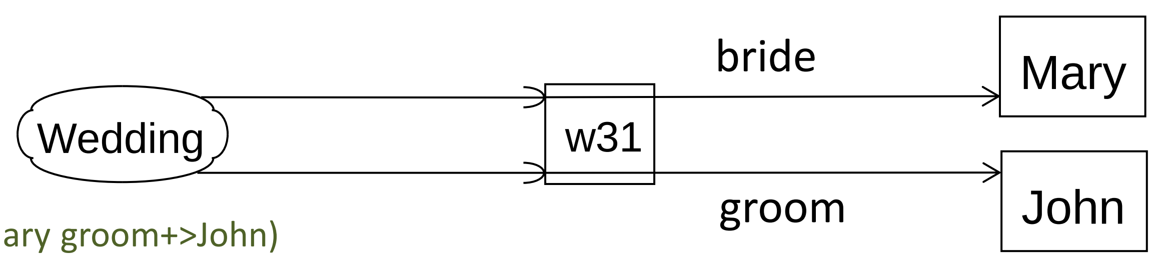
w11#Wedding(+[Mary John])

de3: pairships



Wedding(bride+>Mary groom+>John)

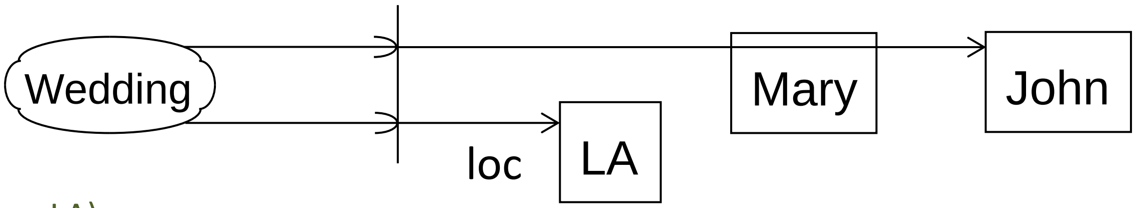
de4. pairpoints



w31#Wedding(bride+>Mary groom+>John)

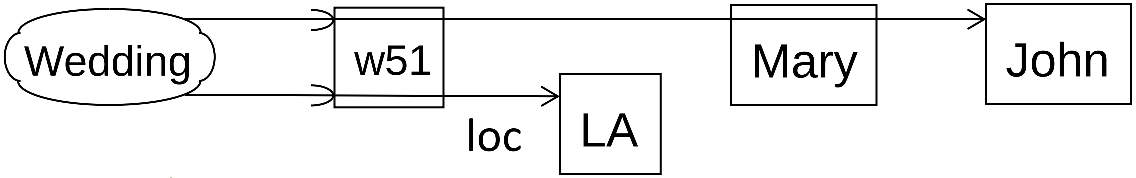
Extra oidless/oidful, combined tupled+slotted atoms that are **de**pendent:

de5. single-tuple:
relpairships



Wedding(+[Mary John] loc+>LA)

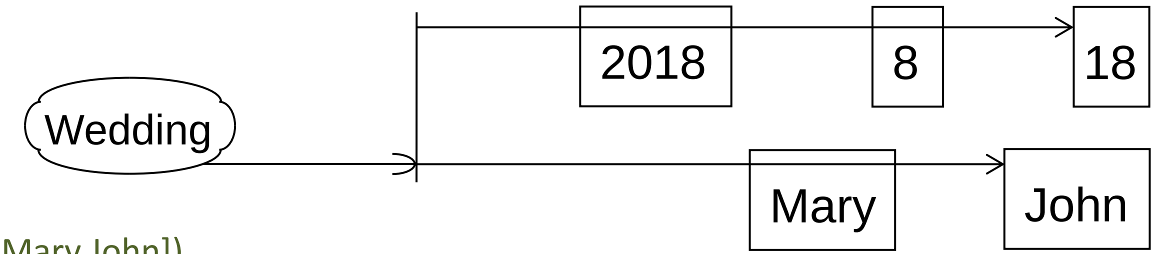
de6. single-tuple:
relpairpoints



w51#Wedding(+[Mary John] loc+>LA)

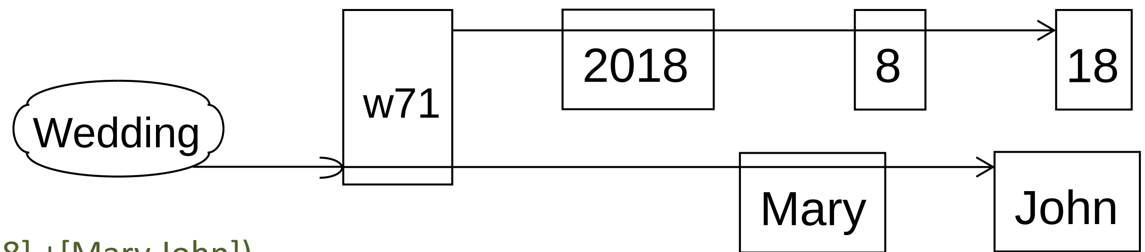
Adding oidless/oidful, tupled/slotted, combined **i**ndependent+**d**ependent atoms:

id1



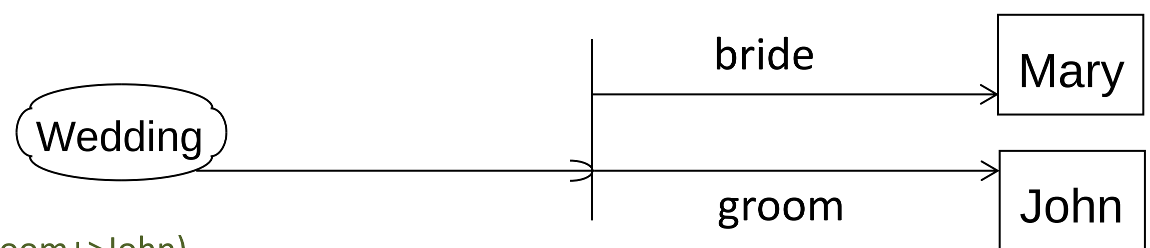
Wedding(-[2018 8 18] +[Mary John])

id2



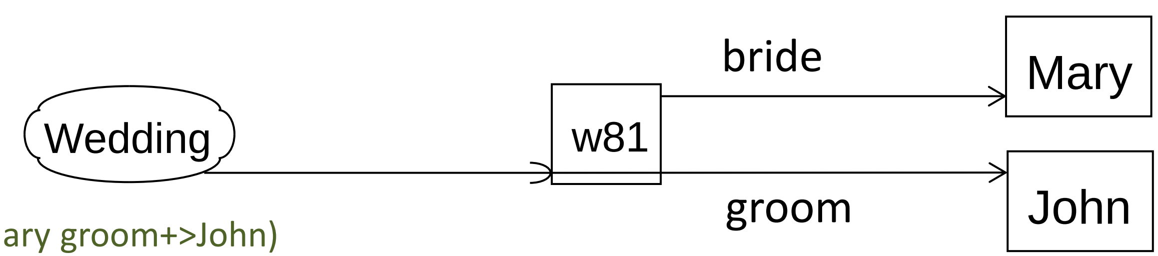
w71#Wedding(-[2018 8 18] +[Mary John])

id3



Wedding(bride->Mary groom+>John)

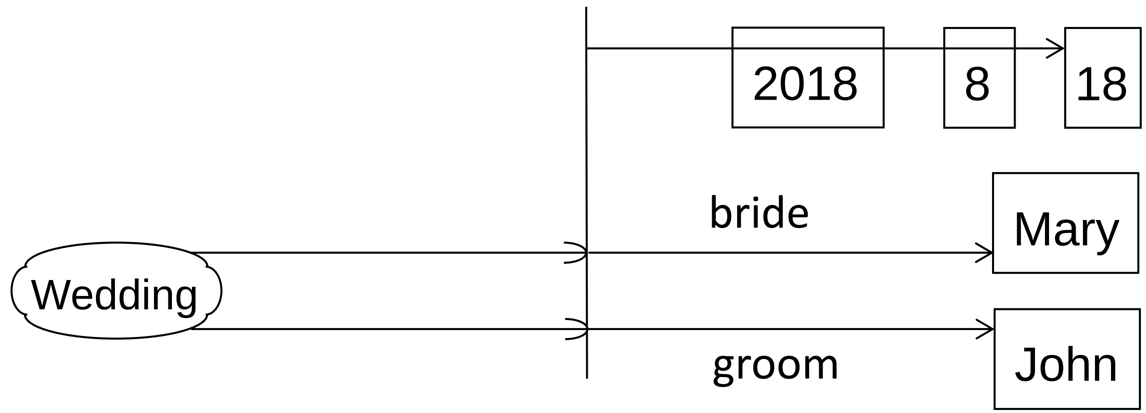
id4



w81#Wedding(bride->Mary groom+>John)

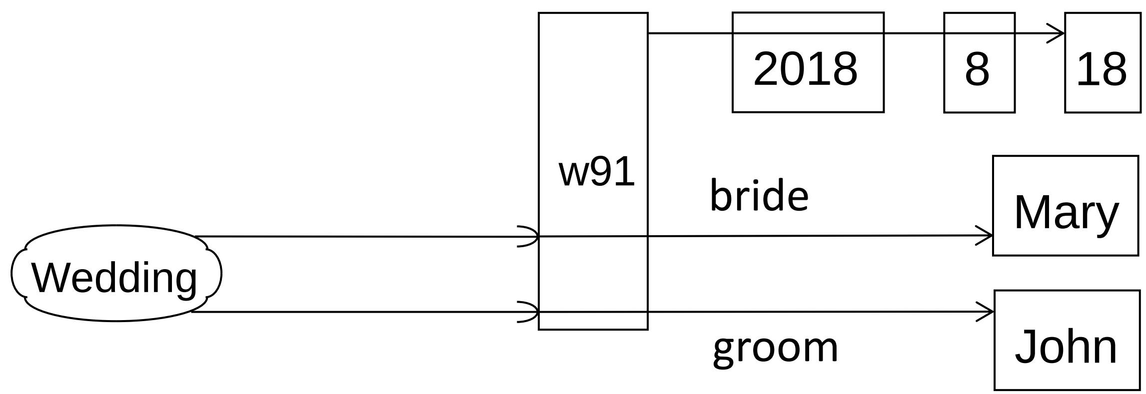
Also oidless/oidful, combined tupled+slotted, combined **i**ndependent+**d**ependent:

id5



Wedding(-[2018 8 18] bride+>Mary groom+>John)

id6



w91#Wedding(-[2018 8 18] bride+>Mary groom+>John)

Syntax and Semantics of Atoms

Core oidless/oidful, tupled/slotted atoms that are **in**dependent:

in1. single-tuple:
shelfships

f(-[t … t] . . . -[t … t]) — Implicit existential OID; tuples -[t … t] independent from predicate f

in2. single-tuple:
shelfpoints

o#f(-[t … t] . . . -[t … t])

o#f(-[t … t]) — Explicit OID o; tuples -[t … t] independent from predicate f

in3: frameships

f(p->v . . . p->v) — Implicit existential OID; slots p->v independent from predicate f

in4: **framepoints**

o#f(p->v . . . p->v) — Explicit OID o; slots p->v independent from predicate f

Extra oidless/oidful, combined tupled+slotted atoms that are **in**dependent:

in5. single-tuple:
shelframeships

f(-[t ... t] . . . -[t ... t] p->v . . . p->v)

Implicit existential OID; descriptors independent from predicate f

in6. single-tuple:
shelframepoints

o#f(-[t ... t] . . . -[t ... t] p->v . . . p->v)

o#f(-[t ... t] p->v . . . p->v)

Explicit OID o; descriptors independent from predicate f

Core oidless/oidful, tupled/slotted atoms that are **de**pendent:

de1. single-tuple:
relationships

f(+[t ... t] . . . +[t ... t])

f(t ... t) *or* f(+[t ... t])

Implicit existential OID; tuples +[t ... t] dependent on predicate f

de2. single-tuple:
relationpoints

o#f(+[t ... t] . . . +[t ... t])

o#f(t ... t) *or* o#f(+[t ... t])

Explicit OID o; tuples +[t ... t] dependent on predicate f

de3: pairships

f(p+>v . . . p+>v)

Implicit existential OID; slots p+>v dependent on predicate f

de4: pairpoints

o#f(p+>v . . . p+>v)

Explicit OID o; slots p+>v dependent on predicate f

Extra oidless/oidful, combined tupled+slotted atoms that are **de**pendent:

de5. single-tuple:
relpairships

f(+[t ... t] . . . +[t ... t] p+>v . . . p+>v)

f(+[t ... t] p+>v . . . p+>v)

Implicit existential OID; descriptors dependent on predicate f

de6. single-tuple:
relpairpoints

o#f(+[t ... t] . . . +[t ... t] p+>v . . . p+>v)   Explicit OID o; descriptors dependent on predicate f

Adding oidless/oidful, tupled/slotted, combined **i**ndependent+**d**ependent atoms:

id1

f(+[t … t] . . . +[t … t]
  -[t … t] . . . -[t … t])

Implicit existential OID; both in/dependent tuples w.r.t. predicate f

id2

o#f(+[t … t] . . . +[t … t]
      -[t … t] . . . -[t … t])

Explicit OID o; both in/dependent tuples w.r.t. predicate f

id3

f(p+>v . . . p+>v
  p->v . . . p->v)

Implicit existential OID; both in/dependent slots w.r.t. predicate f

id4

o#f(p+>v . . . p+>v
      p->v . . . p->v)

Explicit OID o; both in/dependent slots w.r.t. predicate f

Also oidless/oidful, combined tupled+slotted, combined **i**ndependent+**d**ependent:

id5

```
f(+[t ... t] . . . +[t ... t]
  -[t ... t] . . . -[t ... t]
  p+>v . . . p+>v
  p->v . . . p->v)
```

Implicit existential OID; both in/dependent descriptors w.r.t. predicate f

id6

```
o#f(+[t ... t] . . . +[t ... t]
     -[t ... t] . . . -[t ... t]
     p+>v . . . p+>v
     p->v . . . p->v)
```

Explicit OID o; both in/dependent descriptors w.r.t. predicate f

Conclusions

- Full PSOA metamodel cube visualized dynamically by PSOAMetaViz, and atoms (e.g., data facts) in Grailog, to significantly facilitate learning PSOA RuleML
- Facts complemented by (interoperation) rules:
  http://wiki.ruleml.org/index.php/PSOA_RuleML_Bridges_Graph_and_Relational_Databases
  (includes core interoperation path de1-de3-de4-in4, e.g. abridged to one PSOA rule)
- Core path augmented to roundtrip between wedding atoms:
  http://wiki.ruleml.org/index.php/Exploring_the_PSOA_RuleML_Space_of_Core_Atoms
- PSOA RuleML 1.03 being standardized by Relax NG schemas for XML-serialized facts and rules:
  http://wiki.ruleml.org/index.php/PSOA_RuleML#Syntaxes
- PSOA metamodel transferrable to other languages