From RuleML Wiki

PSOATransRun: Translating and Running PSOA RuleML
via the TPTP Interchange Language for Theorem Provers
[1]Faculty of Computer Science,
University of New Brunswick, Fredericton, Canada
[2]National Research Council Canada,
Information and Communications Technologies
University of New Brunswick, Saint John, Canada

Also see documentation, implementation, and sources:
Project Repository: http://psoa2tptp.googlecode.com

For the online demo of the above PSOATransRun:
we give examples of PSOA RuleML Rulebases and Queries,
shown in PSOA Presentation Syntax and Controlled English

Example 1:
Input Rulebase:
Document(
  Group(
    _Amy # _person([_married] [_bcs _mcs _phd] _job->_engineer)
    _f1 # _family(_Mike _Amy _child->_Fred _child->_Jane)
  )
)

Controlled English explaining Fact 1:
_Amy # _person([_married] [_bcs _mcs _phd] _job->_engineer)
(1) The entity with OID _Amy is of type _person;
(2) _Amy is described by:
   a 1-tuple [_married]                representing her marital status
   a 3-tuple [_bcs _mcs _phd]          representing the degrees she obtained
   a _job slot with value _engineer    representing her profession

Controlled English explaining Fact 2:
_f1 # _family(_Mike _Amy _child->_Fred _child->_Jane)
(1) The entity with OID _f1 is of type _family;
(2) _f1 is described by:
   a 2-tuple [_Mike _Amy]              representing the couple of the family
   a _child slot with value _Fred      representing a child of the family
   a _child slot with value _Jane      representing another child of the family

Query 1 with controlled English:
_Amy # _person(_job->_engineer)
_is_Amy of type _person and has a _job slot with value _engineer?

Query 2 with controlled English:
_Amy # _person(_job->?Job)
_is_Amy of type _person and has a _job slot with value ?Job
(show any bindings for ?Job)?
Example 2:
Input Rulebase:

Document (  
  Group (  
    Forall ?X ?Y (  
    )  
    Forall ?X ?Y ?Z (  
      ?X # _person(_descendent->?Z) :-  
        And(?X # _person(_child->?Y) ?Y # _person(_descendent->?Z))  
    )  
    _Tom # _person(_male _child->_Amy _job->_professor)  
    _Eva # _person(_female _child->_Amy)  
    _Amy # _person([_female] [bcs _mcs _phd] _child->_Fred)  
    _Fred # _person(_school->_UNB)  
  )  
)

Controlled English explaining Rule 1:

Forall ?X ?Y (  
  ?X # _person(_descendent->?Y) :-  
  ?X # _person(_child->?Y)  
)  
A _person ?X has a _descendent slot with value ?Y if  
the _person ?X has a _child slot with value ?Y

Controlled English explaining Rule 2:

Forall ?X ?Y ?Z (  
  ?X # _person(_descendent->?Z) :-  
    And(?X # _person(_child->?Y) ?Y # _person(_descendent->?Z))  
)  
A person ?X has a _descendent ?Z if  
the _person ?X has a _child ?Y  
and the _person ?Y has a _descendent ?Z

Query 1 with controlled English:

?Ancestor # _person(_descendent->?Who)  
is ?Ancestor of type _person and has a _descendent slot with value ?Who  
(show any bindings for ?Ancestor and ?Who) ?

Query 2 with controlled English:

And (?Ancestor1 # _person(_male _descendent->_Fred)  
  ?Ancestor2 # _person(_female _descendent->_Fred))  
is ?Ancestor1 of type _person and has a 1-tuple [_male] and a _descendent slot with value _Fred,  
and ?Ancestor2 of type _person and has a 1-tuple [_female] and a _descendent slot with value _Fred  
(show any bindings for ?Ancestor1 and ?Ancestor2) ?

Query 3 with controlled English:

And (?Ancestor1 # _person(_male _descendent->?Who)  
  ?Ancestor2 # _person(_female _descendent->?Who))  
is ?Ancestor1 of type _person and has a 1-tuple [_male] and a _descendent slot with value ?Who,  
and ?Ancestor2 of type _person and has a 1-tuple [_female] and a _descendent slot with value ?Who  
(show any bindings for ?Ancestor1, ?Ancestor2, and ?Who) ?


- This page was last modified on 21 September 2012, at 20:31.