PSOA RuleML Integration of Relational and Object-Centered Geospatial Data

*The 9th International Web Rule Symposium*

*RuleML 2015 Challenge*

*August 2-5, 2015*

Gen Zou

Faculty of Computer Science,
University of New Brunswick, Fredericton, Canada
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Geospatial data sets have been increasingly available on the Web, e.g., Geonames and LinkedGeoData.

Many real-world applications are built on top of local data sets that contain geospatial information.

Integration of application data with external geospatial data can answer interesting geospatial queries.
Data can be modeled in different paradigms

- **Relational**
  - Widely used for relational DBs and KBs, representing information in classical logic

- **Object-centered**
  - Each object is represented by a unique Object IDentifier (OID) typed by a class and described by an unordered collection of slots, each being a pair of a name and a filler

- **Combined**

Integration needs cross-paradigm transformation, which can be expressed in the object-relational rule language PSOA RuleML
PSOA RuleML

- Integrates relational and object-centered modeling
- Generalizes F-logic, RIF-BLD, and POSL
- Uses positional-slotted object-applicative (psoa) terms, permitting a relation application to have an OID – typed by the relation – and, orthogonally, its arguments to be positional or slotted

**General case (multi-tuple):**

\[ o \# f([t_{1,1} \ldots t_{1,n_1}] \ldots [t_{m,1} \ldots t_{m,n_m}] \ p_1->v_1 \ldots p_k->v_k) \]

**Special cases (single-tuple brackets and zero-argument parentheses optional):**

- **Combined:** \[ o \# f([t_1 \ldots t_n] \ p_1->v_1 \ldots p_k->v_k) \]
- **Positional:** \[ o \# f([t_1 \ldots t_n]) \]
- **Slotted:** \[ o \# f( \ p_1->v_1 \ldots p_k->v_k) \]
- **Member-only:** \[ o \# f() \]
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Data Sets

- Two relational data sets and one object-centered data set, expressed in PSOA RuleML presentation syntax

- Relational house rental data set

  ex:HouseRentalInfo(1 "35 Routliffe Lane" "Toronto" "ON" "CA" 3 2500 "False"^^xs:boolean)
  ex:HouseRentalInfo(2 "42 Frey Crescent" "Toronto" "ON" "CA" 2 900 "True"^^xs:boolean)

  Arguments: ref number, street, city, province, country, number of bedrooms, price, furnished
Data Sets

- **Relational data set containing addresses and their GPS coordinates** (From online geocoding services)

  ```
  gc:Geocode(43.778267 -79.426723
  "35 Routliffe Lane" "Toronto" "ON" "CA")
  gc:Geocode(43.742429 -79.291529
  "42 Frey Crescent" "Toronto" "ON" "CA")
  ```

  Arguments: latitude, longitude, street, city, province, country

- **Object-centered data set consisting of geospatial features** (From Geonames)

  ```
  <http://sws.geonames.org/9411373/>#gn:Feature(
  gn:name->"The Detour Store"
  gn:featureCode->gn:S.RET
  geo:lat->45.39748
  geo:long->-80.2468)
  ```
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Hierarchy of Geospatial Entities

- gr:SubwayStation##gr:GeoEntity
- gr:Restaurant##gr:GeoEntity
- gr:Store##gr:GeoEntity
- gr:House##gr:GeoEntity
- gr:HouseForRent##gr:House

- gr:GeoEntity class denotes all geospatial entities that can be located
- Every gr:GeoEntity-typed object has a slot gr:coord for the precise coordinates of its centroid
Integration Rules

- Map house rental data into objects of `gr:HouseForRent` subclass of `gr:GeoEntity` and extract address information

\[
\text{Forall } ?\text{Key} ?\text{Name} ?\text{Phone} ?\text{Street} ?\text{City} ?\text{Prov} ?\text{Country} ?\text{PostCode} ?\text{Addr} \\
\text{Exists } ?\text{Addr} \\
\text{And}(\text{gr:HouseRentID}(?\text{RefNo}) \# \text{gr:HouseForRent}) \\
\text{?Bedrooms ?Price ?Furnished gr:addr->?Addr) \\
?\text{Addr} \# \text{gr:Address (gr:street->?Street} \\
\text{gr:city->?City} \\
\text{gr:prov->?Prov} \\
\text{gr:country->?Country}) \\
\text{ :- ex:HouseRentalInfo(?\text{RefNo} ?\text{Street} ?\text{City} ?\text{Prov} ?\text{Country} \\
?\text{Bedrooms ?Price ?Furnished})}
\]
Integration Rules

- Enrich each GeoEntity with a gr:coord slot, by retrieving the coordinates from gc:Geocode relation using its address


(?
 ?O#gr:GeoEntity(gr:coord->gr:Point(?Lat ?Long))
:- And(?O#gr:GeoEntity(gr:addr->?Ad)
  ?Ad#gr:Address(gr:street->?Street

)
Integration Rules

Map objects from the object-centered data set into objects of gr:GeoEntity

{
    ?O#gr:GeoEntity(gr:name->?Name
        gr:coord->gr:Point(?Lat ?Long))
    :- ?O#gn:Feature(gn:name->?Name
        geo:lat->?Lat
        geo:long->?Long)
}
Map feature codes in the object-centered data set into corresponding gr:GeoEntity subclass

Forall ?O
(
    ?O#gr:SubwayStation
)

Forall ?O
(
    ?O#gr:Restaurant
)

Forall ?O
(
    ?O#gr:Store
)
Derive a GeoEntity \(?O\) is in an \(?Area\) by composing slot \texttt{gr:coord}\ and \texttt{gr:RCCProperPartOf}\ relation

\[
\text{Forall} \?O \ ?Ad \ ?Pt \ ?Area \\
\quad \{ \\
\quad \quad \?O#\text{gr:GeoEntity}(\text{gr:in->}?Area) \\
\quad \quad \quad \text{:- And(} \\
\quad \quad \quad \quad \?O#\text{gr:GeoEntity}(\text{gr:coord->}?Pt) \\
\quad \quad \quad \quad \text{gr:RCCProperPartOf(?Pt ?Area)} \\
\quad \quad \quad \}) \\
\quad \}
\]
Geospatial Relationship Inference Rules

Derive \texttt{gr:RCCProperPartOf} between a point and a box, defined by its minimum latitude, minimum longitude, maximum latitude, and maximum longitude, through arithmetic computation.

\[
\text{Forall } ?\text{Lat} \ ?\text{Long} \ ?\text{LatMin} \ ?\text{LongMin} \ ?\text{LatMax} \ ?\text{LongMax} ( \\
\text{gr:RCCProperPartOf(gr:Point(?Lat \ ?Long) \\
\text{\hspace{2cm} gr:Box(?LatMin \ ?LongMin ?\text{LatMax} \ ?\text{LongMax})) \\
\hspace{2cm} :- \ And ( \\
\text{\hspace{4cm} External(pred:numeric-greater-than-or-equal(?Lat \ ?\text{LatMin})) \\
\text{\hspace{4cm} External(pred:numeric-greater-than-or-equal(?Long \ ?LongMin))) \\
\text{\hspace{4cm} External(pred:numeric-less-than-or-equal(?Lat \ ?\text{LatMax})) \\
\text{\hspace{4cm} External(pred:numeric-less-than-or-equal(?Long \ ?\text{LongMax}))}) \\
\))}
\]
Derive the distance (measured in km) of \(?O_1\) and \(?O_2\) to be less or equal than \(?Distance\), using external function \(\text{gr:distanceLessEqual}\).

\[
\text{Forall } ?\text{Lat1} \ ?\text{Long1} \ ?\text{Lat2} \ ?\text{Long2} \ ?\text{Distance} \ ?\text{Name} \ ?\text{G} \ ?\text{F} \\
\quad ( \\
\quad \quad \text{gr:inDistance}(\?O_1 \ ?O_2 \ ?Distance) \\
\quad \quad : - \\
\quad \quad \quad \text{And}( \\
\quad \quad \quad \quad ?O_1\#\text{gr:GeoEntity}(\text{gr:coord->gr:Point}(\?Lat1 \ ?Long1)) \\
\quad \quad \quad \quad ?O_2\#\text{gr:GeoEntity}(\text{gr:coord->gr:Point}(\?Lat2 \ ?Long2)) \\
\quad \quad \quad \quad \text{External}( \\
\quad \quad \quad \quad \quad \quad \text{gr:distanceLessEqual}(\?Lat1 \ ?Long1 \ ?Lat2 \ ?Long2 \ ?Distance))) \\
\)
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Look for certain type of geospatial entities in a region and their addresses

And(?H#gr:HouseForRent(gr:in->gr:Box(43 -80 44 -79) gr:addr->?Addr)
  ?Addr#gr:Address(gr:street->?Street))

Look for all geospatial entities near specific entities

- All stores within 5km of the house with reference number 2:
  And(?S#gr:Store(gr:name->?Name)
    gr:inDistance(gr:HouseRentID(2) ?S 5))

- All houses within 2km of a subway station and the name of the station
  And(?S#gr:SubwayStation(gr:name->?Name)
    ?H#gr:HouseForRent gr:inDistance(?H ?S 2))
Outline

1. Background
2. Data Sets
3. Rules
4. Queries
5. Conclusion and Future Work
Conclusion and Future Work

- Demonstrate the usefulness of PSOA rules for the integration of geospatial data modeled in different paradigms
- Similar approach can be applied to enrich other local data sets containing address information
- Future work
  - Expand KB with required ground facts imported from relational/graph databases
  - Evaluate reasoning performance on expanded KB using PSOATransRun engine