Trajectory ontology inference over domain and temporal rules

Inference complexity analyzing

Rouaa WANNOUS
rwanno01@univ-lr.fr

Alain BOUJU, Jamal MALKI, Cécile VINCENT

Rouen, July 01, 2014
Outline

1. Context
2. Data modeling
3. Application model
4. Implementation
5. Ontology inference
6. Refinement
7. Conclusion

Trajectory ontology inference over domain and temporal rules
- Mobile object movements
- Capture trajectories

Trajectory ontology inference over domain and temporal rules
Request on moving object trajectory

We aim at answering queries, such as:

1. activities;
2. activities after a given time;
3. in which zones is the moving object doing a specific activity.
Request on moving object trajectory

We aim at answering queries, such as:

1. activities;
2. activities after a given time;
3. in which zones is the moving object doing a specific activity.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Domain</th>
<th>Moving object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Temporal interval</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rules</th>
<th>Domain</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Interval after</td>
</tr>
</tbody>
</table>

Query 2: domain and temporal concepts and rules
Trajectory modeling

Trajectory ontology inference over domain and temporal rules
Trajectory modeling

Moving Object Domain Ontology

Semantic Annotations Ontology

Trajectory Domain Ontology

Trajectory ontology inference over domain and temporal rules
Temporal modeling

W3C OWL-Time ontology

Trajectory ontology inference over domain and temporal rules
Trajectory and time ontologies mapping

Time Ontology

Instant

e_date = hasEnd

hasEnd

hasBeginning

 ProperInterval

owl:equivalentClass

= is owl:equivalentProperty

Trajectory Ontology

Thing

rdfs:subClass

Trajectory

rdfs:subClass

Sequence

rdfs:subClass

ProperInterval

Instant

e_date = hasEnd

s_date = hasBeginning

hasBeginning

hasEnd

Instant

ProperInterval

owl:equivalentClass

= is owl:equivalentProperty

Trajectory ontology inference over domain and temporal rules
### Application scenario

**Trajectory ontology inference over domain and temporal rules**
State of seal trajectory

Seal states

- Haulout: continuously wet for 40 sec
- Cruise: continuously dry for 10 mins, shallower than 1.5m
- Dive: deeper than 1.5m for 8 secs

Sensor deployment start date time is known
Sensor deployment end date time is known
Seal trajectory ontology
General framework implementation

We are based on RDF triple store:

1. create declarative parts of ontologies;
2. populate the domain and the time ontologies;
3. create imperative parts of ontologies (domain trajectory activities and temporal ontology rules)
Seal trajectory ontology rules

<table>
<thead>
<tr>
<th>Rules</th>
<th>Maximum dive depth (meter)</th>
<th>Dive shape or TAD</th>
<th>Surface ratio = surface dur/dive dur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resting</td>
<td>&lt; 10</td>
<td>all</td>
<td>&gt; 0.5</td>
</tr>
<tr>
<td>Travelling</td>
<td>&gt; 3</td>
<td>&gt; 0 &amp; &lt; 0.7</td>
<td>&lt; 0.5</td>
</tr>
<tr>
<td>Foraging</td>
<td>&gt; 3</td>
<td>&gt; 0.9 &amp; &lt; 1</td>
<td>&lt; 0.5</td>
</tr>
<tr>
<td>Travelling Föraging</td>
<td>&gt; 3</td>
<td>&gt; 0.7 &amp; &lt; 0.9</td>
<td>&lt; 0.5</td>
</tr>
</tbody>
</table>

Decision table associated to seal activities

*TAD: Time Allocation at Depth, shape of dives*
Seal trajectory ontology rules

```
1 rdf:type(?diveObject, s:Dive) ^
2 s:max_depth (?diveObject, ?maxDepth) ^
3 s:tad (?diveObject, ?diveTAD) ^
4 s:dive_dur (?diveObject, ?diveDur) ^
5 s:surf_dur (?diveObject, ?surfaceDur) ^
6 s:seqHasActivity (?diveObject, ?activityProperty) ^
7 > (maxDepth, 3) ^
8 > (diveTAD, 0.9) ^
9 < (surfaceDur/diveDur, 0.5) ^
10 ==> 
11 rdf:type (?activityProperty, s:Foraging)
```

The imperative part of the seal activity foraging
Time ontology rules

Trajectory ontology inference over domain and temporal rules
Time ontology rules

13 Allen temporal rules

Before 1

Overlap 1

Starts 2

Equal 2

Meets 1

During 2

Finishes 2
IntervalAfter rule

1. \( \text{rdf:type(?timeObject1, owltime:ProperInterval)} \)
2. \( \text{owltime:hasEnd(?timeObject1, ?EndDateTime1)} \)
3. \( \text{rdf:type(?timeObject2, owltime:ProperInterval)} \)
4. \( \text{owltime:hasBeginning(?timeObject2, ?BeginDateTime2)} \)
5. \( > (\text{BeginDateTime2, EndDateTime1}) \)
6. \( \implies \)
7. \( \text{owltime:intervalAfter(?timeObject2, ?timeObject1)} \)

The imperative part of Allen temporal relationship intervalAfter
Ontology inference entailment over trajectory and temporal rules

```java
SEM_APIs.CREATE_ENTAILMENT('owlSealTrajectory_idx',
SEM_MODELS('owlSealTrajectory', 'owlTime'),
SEM_RULEBASES('OWLPrime', 'sealActivities_rb', 'owlTime_rb'),
SEM_APIs.REACH_CLOSURE, NULL, 'USER_RULES=T');
```

Ontology inference entailment over trajectory and temporal rules
Experiments over temporal ontology inference

![Temporal rules graph]

- Time x 10,000 seconds
- Number of dives

Temporal rules

Rouaa WANNOUS

Trajectory ontology inference over domain and temporal rules
## Ontology inference

### Demo

Trajectory ontology inference over domain and temporal rules
Tow-tier inference filters

raw data → Primary inference → Place-Of-Interest

rules → Secondary inference

Knowledge base → query

data

Trajectory ontology inference over domain and temporal rules
Interesting places and foraging places

Trajectory ontology inference over domain and temporal rules
Conclusion

1. Trajectories lack semantic information
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology
2. Semantic trajectories need other sources
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology

2. Semantic trajectories need other sources
   - Integrate time ontology and temporal relationships

Trajectory ontology inference over domain and temporal rules
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology
2. Semantic trajectories need other sources
   - Integrate time ontology and temporal relationships
3. Complexity of computing the inference mechanism

Trajectory ontology inference over domain and temporal rules
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology

2. Semantic trajectories need other sources
   - Integrate time ontology and temporal relationships

3. Complexity of computing the inference mechanism
   - Define two-tier inference filters
Conclusion

1. Trajectories lack semantic information
   - Built semantic trajectory ontology

2. Semantic trajectories need other sources
   - Integrate time ontology and temporal relationships

3. Complexity of computing the inference mechanism
   - Define two-tier inference filters
   - Verify the positive impact of the proposal on reducing the complexity of the inference mechanism on real GPS data