Storing and Querying the Valid Time of Triples in Linked Geospatial Data

Konstantina Bereta, Panayiotis Smeros, Manolis Koubarakis

Konstantina.Bereta@di.uoa.gr

ESWC 2013, Montpellier

Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece
Outline

• **Introduction**

• The “t” of stRDF and stSPARQL

• Implementation in Strabon

• Experimental evaluation

• Future Work and Conclusions
Time dimensions in linked geospatial data

- **User-defined time**: A time value (literal) with no special semantics.

- **Valid time**: The time when a fact (represented by a triple) is true in the modeled reality.

- **Transaction time**: The time when the triple is current in the database.
Example of valid time
Previous work

• **Time in relational databases**
  - TSQL2 (1995)
  - SQL:2011 (Oracle Workspace manager, IBM DB2, Teradata)

• **Time in RDF**
  - Temporal RDF (Gutierrez et al., ESWC 2005)
  - Annotated RDFS (Lopes et al., ISWC 2010)
  - Mapping to standard RDF:
    - Reification
    - Named graphs (Tappolet and Bernstein, ESWC 2009)
Outline

• Introduction

• The “t” of stRDF and stSPARQL
  • Implementation in Strabon
  • Experimental evaluation

• Future Work and Conclusions
The “s” in stRDF and stSPARQL

- **stRDF**: Extends RDF for representing geospatial data
  - **Objects in triples can be spatial literals** encoded in Well-Known Text/GML (OGC standards)

- **stSPARQL**: Extends SPARQL 1.1 with **spatial extension functions** defined in OGC standards for querying geospatial data.
Strabon (http://strabon.di.uoa.gr)

- WKT
- GML
- stRDF graphs
- stSPARQL/GeoSPARQL queries

Strabon

Query Engine:
- Parser
- Optimizer
- Evaluator
- Transaction Manager

Storage Manager:
- Repository
- SAIL
- RDBMS

GeneralDB

PostGIS

monetdb
The “t” in stRDF

- **User-defined time:** already supported in RDF
- **Valid time:** Triples are annotated by their valid time
  - Triples are extended to **quads** (the last component is the valid time of the triple)
  - **Instants** represented using **xsd:dateTime** values
  - **Periods** represented using literals of the new datatype **strdf:period**
  - **Temporal Constants:** **NOW**, **UC(“Until Changed”)**
The “t” in stSPARQL

SPARQL 1.1 is extended as follows:

• **Triple patterns** extended to **quad patterns**
  (the last component is a temporal term)

• **Temporal extension functions**
  • Allen's temporal relations (e.g., before, meets, overlaps)
  • Nine more functions that are syntactic sugar to the above
    (e.g., intersects, startsWithinOrAfter, finishesWithinOrBefore)
  • Interval-to-point relations (e.g., during, before, after)
  • Period constructors (e.g., period_union, period_intersect)
  • Temporal aggregates (e.g., maximalPeriod, intersectAll)
  • Temporal Updates

• Defined under strdf namespace: [http://strdf.di.uoa.gr/ontology](http://strdf.di.uoa.gr/ontology)
Example
Example stRDF graph

clc:region1 rdf:type clc:Region .
clc:region1 strdf:hasGeometry "POLYGON((23.38 40.48, ... 23.38 40.48))"^^strdf:WKT .

clc:region1 clc:hasLandCover clc:Forest "[2006-08-25T11:00:00+02, 2007-08-25T11:00:00+02)"^^strdf:period .

noa:ba1 rdf:type noa:BurntArea "[2007-08-25T11:00:00+02, 2009-08-25T11:00:00+02)"^^strdf:period .

noa:ba1 strdf:hasGeometry "POLYGON((23.26 40.51, ... 23.26 0.51))"^^strdf:WKT.

clc:region1 clc:hasLandCover clc:AgriculturalArea "[2009-08-25T11:00:00+02, "UC")"^^strdf:period .
stSPARQL query (1/2)

• Find the **current** land cover of all areas in the dataset

```sparql
SELECT ?clc
WHERE {
  ?R rdf:type clc:Region .
  FILTER(strdf:during ("NOW", ?t1))
}
```

Quad Pattern

Temporal constant

Temporal extension function
Find all areas that were first covered by a forest, then got burnt, and then became agricultural areas.

SELECT ?R

WHERE {
  ?R rdf:type clc:Region .

  FILTER(strdf:after(?t2, ?t1)) .

  FILTER(strdf:after(?t3, ?t2)) }

Temporal extension function
Outline

• Introduction

• The “t” of stRDF and stSPARQL

• Implementation in Strabon

• Experimental evaluation

• Future Work and Conclusions
Translating quads to triples

- Example:

```
clc:region1 clc:hasLandCover clc:Forest

"[2006-08-25T11:00:00+02, 2007-25T11:00:00+02)"^^strdf:period.
```

GRAPH strdf:validTimeURI_1

```
clc:region1 clc:hasLandCover clc:Forest
```

DEFAULT GRAPH

```
strdf:validTimeURI_1 strdf:hasValidTime

"[2006-08-25T11:00:00+02, 2007-25T11:00:00+02)"^^strdf:period
```

Translating quad patterns to triple patterns

Example

SELECT ?clc

WHERE {?R rdf:type clc:Region .

FILTER(strdf:during ("NOW", ?t1))}

SELECT ?clc

WHERE {?R rdf:type clc:Region .

GRAPH ?g {?R clc:hasLandCover ?clc .}
?q strdf:hasValidTime ?t1 .
FILTER(strdf:during ("NOW", ?t1))}
Strabon extended with valid time
• Introduction

• The “t” of stRDF and stSPARQL

• Implementation in Strabon

• **Experimental evaluation**

• Future Work and Conclusions
Competitor Systems

• Baseline implementation: Sesame Native Store

• Prolog-based implementation of AnQL (Lopes et al.)

• AllegroGraph (Free server edition v4.10)
Datasets

- GovTrack dataset
  - 8M triples
  - 42K periods
  - 294K instants

- Corine Land Cover changesets 2000-2006
  - 795K triples
  - 718K periods
Query response time as input dataset grows

Cold caches

Warm caches

Time (milliseconds)

Number of triples and quads
## Query response time with respect to query complexity – cold caches (sec)

<table>
<thead>
<tr>
<th>System</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strabon</td>
<td>0.3</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
<td>0.7</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.9</td>
<td>2.1</td>
<td>2.2</td>
<td>2.3</td>
<td>2.3</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>AnQL</td>
<td>2.0</td>
<td>1.7</td>
<td>4.2</td>
<td>4.3</td>
<td>5.8</td>
<td>6.9</td>
<td>7.4</td>
</tr>
<tr>
<td>AllegroGraph</td>
<td>1.0</td>
<td>58.1</td>
<td>91.7</td>
<td>121.8</td>
<td>154.6</td>
<td>33.8</td>
<td>156.4</td>
</tr>
<tr>
<td>Results</td>
<td>12203</td>
<td>781</td>
<td>781</td>
<td>781</td>
<td>781</td>
<td>128</td>
<td>87</td>
</tr>
</tbody>
</table>
Query response time with respect to query complexity - warm caches (sec)

<table>
<thead>
<tr>
<th>System</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strabon</strong></td>
<td>0.01</td>
<td>0.07</td>
<td>0.9</td>
<td>0.1</td>
<td>0.1</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>0.7</td>
<td>1.6</td>
<td>1.7</td>
<td>1.7</td>
<td>1.8</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>AnQL</strong></td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.3</td>
<td>0.9</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>AllegroGraph</strong></td>
<td>1.0</td>
<td>56.3</td>
<td>90.8</td>
<td>123.5</td>
<td>154.6</td>
<td>157.8</td>
<td>158.0</td>
</tr>
<tr>
<td><strong>Results</strong></td>
<td>12203</td>
<td>781</td>
<td>781</td>
<td>781</td>
<td>781</td>
<td>128</td>
<td>87</td>
</tr>
</tbody>
</table>
Outline

• Introduction

• The “t” of stRDF and stSPARQL

• Implementation in Strabon

• Experimental evaluation

• **Future Work and Conclusions**
Conclusions and Future work

Conclusions

• We defined the valid time dimension of stSPARQL

• Strabon is a very rich spatio-temporal RDF store, offering efficient evaluation of temporal operators

Future work

• Experiments with even larger datasets, different storage schemes and query evaluation strategies

• Continue systematic evaluation of temporal RDF stores

• Representation and querying of moving objects
Thank you for your attention!

Questions?

- **Strabon** ([http://strabon.di.uoa.gr](http://strabon.di.uoa.gr))
  - Mercurial repository: [http://hg.strabon.di.uoa.gr](http://hg.strabon.di.uoa.gr)
- Sextant: A web tool for browsing and mapping Linked Geospatial Data ([http://test.strabon.di.uoa.gr/sextant/](http://test.strabon.di.uoa.gr/sextant/)) (visit our demo!)
- TELEIOS EU Project ([http://www.earthobservatory.eu](http://www.earthobservatory.eu)) (EU Project NW session)