A Query Model for Ontology Based Event Processing on RDF Streams

DEBS - 26.06.2019
Darmstadt, Germany, Europe, Earth, Milky Way, Universe......42

Pieter Bonte, Riccardo Tommasini, Filip De Turck, Femke Ongenae, Emanuele Della Valle
Authors
That Asks Complex Questions

Who is driving event in the news right now?

Is public transportation where the people are?

Who are the best agents to route all these unexpected contacts about the tariff plan launched yesterday?

What is the expected time to failure when a turbine's barring starts to vibrate as detected in the last 10 minutes?
Enrich the Streams
Crossing the Streams
Issues

• Need to manually register network of queries that produced the complex results.

• Need to manually perform data integration adapting sources and hard-code their semantics.

• Need to maintain such complex systems once you have it!
Is it possible to make sense in real time of multiple, heterogeneous, gigantic and inevitably noisy and incomplete data streams in order to support the decision processes of extremely large numbers of concurrent users?

E. Della Valle, S. Ceri, F. van Harmelen & H. Stuckenschmidt, 2010
Stream Reasoning

Is there a primary cool colour followed by a secondary warm one in the last minute?

Which are the top-2 most frequent cool colours in the last minute?

1 minute wide window

An ontology of
Stream Reasoning

Which are the most frequent **sentiments** in the last minute?

Is there an *impulsive*, *irritating* colour followed by a **calm** one in the last minute? Yes, __followed by__.

1 minute wide window

---

**A different ontology (of colours)**

- intense, fierce, love, anger, excitement, strength, irritating, lips, hearts, sexy, romance, sensuality, impulsive, leadership, courage, competence, independence, organization, self-motivation, spirituality, pleasure, vitality, will to win, survival, instinct, intuition, entrepreneurial, desire, fire, stimulation, joy, rage, sunshine, tropical, enthusiasm, fascination, happiness, creativity, attraction, success, citrus, endurance, illumination, wisdom, wealth, intellect, loyalty, freshness, growth, harmony, fertility, safety, money, vision, experience, novice, hope, nature, finance, ambition, greed, jealousy, healing, protection, peace, sky, sea, depth, trust, confidence, faith, truth, heaven, mind, tranquility, calm, sincerity, clean, water, mineral, precision, expertise, understanding, softness, knowledge, power, royalty, nobility, luxury, extravagance, dignity, mystery, magic, artificial, nostalgia, gloom, frustration, light, goodness, innocence, purity, perfection, positive, beginning, cool, simplicity, charity, angels, sterility, elegance, formality, evil, fear, unknown, feeling, authority, pride, grief.
Stream Reasoning
(in practice)

- Addresses data variety employing semantic technologies,
  - RDF, SPARQL, and reasoning methods (materialisation, or query rewriting)

- Addresses data velocity employing stream processing technologies
  - Data Stream Management Systems, Complex Event Processing, and Big Streaming Systems (check out our tutorial)
Stream Processing

Semantic Web
Research Question

• (Mezzo) Leverage advanced language features in streaming systems to speed up throughput of stream reasoning problems

• (micro) Leveraging inheritance in streaming languages to speed up hierarchical stream reasoning
Contributions

- C-TRA, an extension of Taxonomy-Relational Algebra [20] to the continuous semantics
- Alignment of query answering in CTRA to query answering in (continuous) SPARQL under RDFS entailment regime
- CSPRITE: two algorithms to efficient hierarchical reasoning of RDF streams.
h-domain

- Nation
  - Italy
  - UK
  - Germany

- City
  - Milan
  - London
  - Darmstadt

- Place
  - San Siro
  - Stamford Bridge
  - Wembley
  - Darmstadtium

Mapping Relation
taxonomy

Nation
City
Place

Italy
UK
Germany

Milan
London
Darmstadt

San Siro
Stamford Bridge
Wembley
Darmstadtium
T-Schema & T-Relation

Type table

<table>
<thead>
<tr>
<th>Subject</th>
<th>obs:l2</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_a</td>
<td>obs_x WeatherObservation</td>
</tr>
<tr>
<td>t_b</td>
<td>obs_y GeometryObservation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subject</th>
<th>obs:l2</th>
<th>obs:l1</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_a</td>
<td>obs_x WeatherObservation</td>
<td>Observation</td>
</tr>
<tr>
<td>t_b</td>
<td>obs_y GeometryObservation</td>
<td>Observation</td>
</tr>
</tbody>
</table>

upward extension
The CQL model

Streams

Relations

Relation R(t)

Mapping: T \rightarrow R

Sliding windows

stream-to-relation

relation-to-stream

Relation algebra

relation-to-relation

Stream operators

finite bag

infinite unbounded sequence

<s, T>

<s_1>

<s_2>

<s_3>
The CTQL model

Streams

T-Relations

Sliding windows

stream-to-relation

relation-to-stream

*Stream operators

finite bag

Mapping: T \rightarrow R

t-Relation R(t)

TRA

t-relation-to-t-relation

infinite unbounded sequence

<s,t>

<s1>

<s2>

<s3>

Stream
The CTQL model

Streams

- infinite unbounded sequence
- \( t-s_1 \)
- \( t-s_2 \)
- \( t-s_3 \)

Sliding windows

Stream operators

Relations

Relation-to-relation

\( R(t) \) - Mapping: \( T \rightarrow R \)

Relation-to-stream

Stream-to-relation

Relational Algebra
RDF and RDFS Ontologies
TRA and RDFS Ontologies
TRA and RDFS Ontologies
RDF Stream Processing UE

Q1: ?w a CreativeWork

a1 a Article
a1 a WrittenWork
a1 a CreativeWork

Materialisation!
RDF Stream Processing UE

Q1: ?w a CreativeWork
Q2: ?w a Software
Q3: ?w a VideoGame
Q4: ?w a Article
Q5: ?w a Book
Q6: ?w a WrittenWork

a1 a Article

subClassOf

Concept

Work

CreativeWork

Software

WrittenWork

VideoGame

Article

Book
Algorithm 1 Query registering

**Precondition:** $Q$ a collections of queries, each interested in one or more types.

1. $H \leftarrow \textit{ConvertToHierarchy}(O)$ \Comment{Stores parents for each class in the Ontology $O$}
2. function \textit{PrepareHierarchy($H, Q$)}
3. \hspace{1em} $H' \leftarrow []$
4. \hspace{1em} for $q \in Q$ do
5. \hspace{2em} for $(\text{concept}, \text{parents}) \in H$ do
6. \hspace{3em} if $q \in \text{parents}$ then
7. \hspace{4em} $H'[\text{concept}].\text{append}(q)$
8. \hspace{2em} end if
9. \hspace{1em} end for
10. end for
11. return $H'$
12. end function

Algorithm 2 Calculate the query matches on a hierarchical level

**Precondition:** $Q$ a collections of queries, each interested in one or more types.

1. $H \leftarrow \textit{ConvertToHierarchy}(O)$ \Comment{Stores parents for each class in the Ontology $O$ (preprocessing step)}
2. $H' \leftarrow \textit{PrepareHierarchy}(H, Q)$ \Comment{(preprocessing step)}
3. $\text{triple} \leftarrow \textit{ClassAssertion}(\text{type,subject})$
4. function \textit{CHECKHIERARCHYMATCH($H', \text{triple}$)}
5. \hspace{1em} $\text{QueryMatches} \leftarrow H'(\text{types(triple)})$ \Comment{types extracts the type assertions of a triple}
6. \hspace{1em} return $\text{QueryMatches}$
7. end function
C-Sprite

Concept: [Concept]
Work: [Work, Concept]
CreativeWork: [CreativeWork, Work, Concept]
WrittenWork: [WrittenWork, CreativeWork, Work, Concept]
Article: [Article, WrittenWork, CreativeWork, Work, Concept]
Book: [Book, WrittenWork, CreativeWork, Work, Concept]
Software: [Software, CreativeWork, Work, Concept]
VideoGame: [VideoGame, Software, CreativeWork, Work, Concept]
C-Sprite

Concept: [Concept]
Work: [Work, Concept]
CreativeWork: [CreativeWork, Work, Concept]
WrittenWork: [WrittenWork, CreativeWork, Work, Concept]
Article: [Article, WrittenWork, CreativeWork, Work, Concept]
Book: [Book, WrittenWork, CreativeWork, Work, Concept]
Software: [Software, CreativeWork, Work, Concept]
VideoGame: [VideoGame, Software, CreativeWork, Work, Concept]
C-Sprite

Concept: [Concept]
Work: [Work, Concept]
CreativeWork: [CreativeWork, Work, Concept]
WrittenWork: [WrittenWork, CreativeWork, Work, Concept]
Article: [Article, WrittenWork, CreativeWork, Work, Concept]
Book: [Book, WrittenWork, CreativeWork, Work, Concept]
Software: [Software, CreativeWork, Work, Concept]
VideoGame: [VideoGame, Software, CreativeWork, Work, Concept]
**Evaluation**

Change stream

Q1: ?w a CreativeWork

<table>
<thead>
<tr>
<th>Top 5 Creative Works:</th>
<th>Absolute Number</th>
<th>Relative Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>MusicalWork</td>
<td>21.438</td>
<td>0.61%</td>
</tr>
<tr>
<td>Film</td>
<td>13.890</td>
<td>0.40%</td>
</tr>
<tr>
<td>WrittenWork</td>
<td>6.814</td>
<td>0.19%</td>
</tr>
<tr>
<td>TelevisionShow</td>
<td>4.579</td>
<td>0.13%</td>
</tr>
<tr>
<td>Software</td>
<td>4.493</td>
<td>0.13%</td>
</tr>
<tr>
<td>Creative Works</td>
<td>56.581</td>
<td>1.61%</td>
</tr>
<tr>
<td>all triples</td>
<td>3,511.629</td>
<td>100%</td>
</tr>
</tbody>
</table>
Evaluation: Increasing window size
Evaluation: Increasing ontology depth

![Graph showing maximum throughput increasing ontology depth](image1)

![Graph showing memory usage increasing ontology depth](image2)
Future Work

- TRA @ Apache Spark
- Query Containment
Questions?

- Email: riccardo.tommasini@polimi.it
- Twitter: @rictomm
- Github: riccardotommasini
- Web1: riccardotommasini.com
- Web2: streamreasoning.org
- Web3: streaminglang.io