Towards Efficient Semantically Enriched Complex Event Processing and Pattern Matching

Syed Gillani$^{1,2}$  
Gauthier Picard$^1$  
Frédérique Laforest$^2$  
Antoine Zimmermann$^1$

Institute Henri Fayol, EMSE, Saint-Etienne, France $^1$

Telecom Saint Etienne, Université Jean Monnet, Saint-Etienne, France $^2$
OVERVIEW

Introduction
  Traditional Vs Real-Time Data Processing
  Event Processing Vs Time Axis
  Complex Event Processing

SEMANTIC COMPLEX EVENT PROCESSING

PROPOSED APPROACH

CONCLUSION
TRADITIONAL VS REAL-TIME DATA PROCESSING

Traditional Data Processing
- One Shot Database Queries
- Database
- Query Processor

Real-Time Data Processing
- Incoming Events
- Time-Future
- Time-Current
- Time-Past
- Event Arrival Time
- Continuous Event Query Processing

Traditional Data Processing
- Database
- One Shot Database Queries
- Query Processor

Real-Time Data Processing
- Incoming Events
- Time-Future
- Time-Current
- Time-Past
- Event Arrival Time
- Continuous Event Query Processing
Event Processing Vs Time Axis

*Dr. Adrian Paschke, DemAAL Summer school 2013*
Complex Event Processing

- Aggregation, derivation of Primitive Events
- Occurrence and non-occurrence of certain events
- Imposing Temporal Constraints (application of certain rules)
- For Instance
  - Detection of state changes based on observations (If total consumed electricity > 10MWatt)
  - Matching sequence of events that describes a scenario (If A<10 AND B>40 OR B<80 AND C>90)
OVERVIEW

Introduction

Semantic Complex Event Processing
SCEP
State-of-the-art SCEP
Foundational Challenges for SCEP

Proposed Approach

Conclusion
SCEP

- Complex Event Processing + Stream Reasoning + Semantic Technologies (rules & ontologies) + Heterogeneous Data Handling?

- Incoming Stream Reasoning + Background Knowledge

- Distributed into TWO flavours
  - Stream Reasoning (Real Time + Background Information + Aggregation through Windows) (C-SPARQL, CQELS....)
  - Pattern Matching (Sequence, Optional, Negation) (EP-SPARQL)
# State-of-the-art SCEP

<table>
<thead>
<tr>
<th></th>
<th>Continuous Query</th>
<th>Background Knowledge</th>
<th>Data Model</th>
<th>Event Processing (Per Query)</th>
<th>Historical Data (No Dedicated Management)</th>
<th>Underlying Engine</th>
<th>Parallel and Distributed Multi-Query Processing</th>
<th>Temporal Operators (Pattern Matching)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-SPARQL</td>
<td>✓</td>
<td>✓</td>
<td>Triple Based</td>
<td>Centralised</td>
<td>x</td>
<td>ESPER</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>CQLES</td>
<td>✓</td>
<td>✓</td>
<td>Triple Based</td>
<td>Centralised</td>
<td>x</td>
<td>ESPER</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>EP-SPARQL</td>
<td>✓</td>
<td>✓</td>
<td>Triple Based</td>
<td>Centralised</td>
<td>✓</td>
<td>ETAILS</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Streaming SPARQL</td>
<td>✓</td>
<td>✓</td>
<td>Triple Based</td>
<td>Centralised</td>
<td>x</td>
<td>DYNQUEST</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>TA-SPARQL</td>
<td>✗</td>
<td>✓</td>
<td>Triple Based</td>
<td>Centralised</td>
<td>✓</td>
<td>TUPELO</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

*Streaming the Web: Reasoning over Dynamic Data: Alessandro Margara, Jacopo Urbani, Frank van Harmelen, Henri Bal*
State-of-the-art SCEP

- Complex Pattern Matching (Approaches)
  - Relational Community
    - NFA, EDG, RETE algorithm, Rule based system
  - Semantic Web Community
    - RETE algorithm, Logical Rule based system
- How about NFA and EDG in SCEP context?
- NFA and EDG are proven to be the most efficient for Pattern Matching in relational community

*Non-Deterministic Finite Automata
*Event Detection Graphs
Foundational Challenges for SCEP

- **Distributed Event Processing (per Query):** Moving from centralised push based event processing
- **Distributed Temporal Pattern Matching:** Dedicated language for Pattern Matching (Implementation of Kleene Closure, Negation in distributed manner)
- **Historical Management of Events:** Storing and Partitioning of events
- **Defining Event Boundaries:** Triple based to Graph based streaming, preserving graph model to implement Event boundaries
- **Predictive Event Processing:** A new paradigm for SCEP
- **Stream Reasoning + CEP:** Combing two different worlds
Overview

Introduction

Semantic Complex Event Processing

Proposed Approach
- Event and Stream Data Model
- Query Model and Language Specification

Conclusion
Event and Stream Data Model

- Considering RDF as first class citizen (even for temporal reasoning, instead relying on external engines)
- Temporally Annotated RDF Named Graph
  \(< NG, [ts, te] >\)

\(<http://www.streaminginfo.com/ElecGen> [st1, et1]
  :gen1 :hasName ‘PowGen-Sect1’.
  :gen1 :hasLocation ‘St-Etienne’.
  :gen1 :hasCurrentPower ‘60’.
Proposed Data Model

- Data Partitioning == $\rightarrow$ Optimises query time
- Summarisation == $\rightarrow$ Merging of similar NG
- Event Boundaries == $\rightarrow$ With NG
- Access Control == $\rightarrow$ With NG
- Provenance Tracking == $\rightarrow$ With NG
- Fact Assignment == $\rightarrow$ With Time Interval
QUERY MODEL AND LANGUAGE SPECIFICATION

- Former Query Models
  - Reliance on Triple-Based Data Model
    - Uses black-box approach (delegation to external Engines)
  - Overhead in query and data translation
  - Query Semantics not suitable for distributed processing per query (SPARQL Extensions...)

Former Query Models
**Proposed Query Model**

**Sub-Query 1 (Event Pattern A)**

```sql
PREFIX sm: <http://example.com/sm>
PREFIX lv: <http://example.com/lv>

Select *
Within 12 hours
From Stream S1 <http://example.org/streams/powersource> Window From Now 10 mins
From Stream S2 <http://example.org/streams/weathersource> Window From Now 10 mins
From Stream S3 <http://example.org/streams/elecappliance> Window From Now 10 mins
Where {
SEQ (EVENTPATT A, (EVENTPATT B)+, (EVENTPATT C AND EVENTPATT B))

DEFINE EVENTPATT A ON S1 { ?event rdfs:subclassof owl:thing; sm:events[ sm:eventType sm:powersource; id ?id; sm:powers ?pow]. GRAPH <http://example.org/streams/sourcelocation> {?id lv:name ?locName} FILTER( ?id = 'gen1', ?pow = '60') }


}
```

**Rewritten Subqueries (Stream Processing)**

**Sub-Query 2 (Event Pattern B)**

```sql
Select *
From Stream S2 <http://example.org/streams/weathersource> Window From Now 10 mins
Where {
(?event rdfs:subclassof owl:thing; sm:events[ sm:eventType sm:weathersource; :id ?id; sm:temp ?temp; sm:pressure ?pres].)
FILTER( ?id = 'Wsource1', ?temp = '20', ?pres = '10')
}
```

**Sub-Query 3 (Event Pattern C)**

```sql
Select *
From Stream S3 <http://example.org/streams/elecappliance> Window From Now 10 mins
Where {
(?event rdfs:subclassof owl:thing; sm:events[ sm:eventType sm:elecappliance; name ?name; sm:usagepower ?pow; sm:loadclass ?load].)
FILTER( ?id = 'heater', ?pow < genpow ?load = '10-100Watt')
}
```
SYSTEM OVERVIEW

Stage 1: Stream Selection
- Stream_i: Incoming Streams
- J_i: Join Operations
- E_n: Event Nodes

Stage 2: Continuous Query Processing and Inference
- S_k: Select Operators
- G_t: Generated Events
- A/B/D: NFA States

Stage 3: Rule or Pattern Mapping
- Pattern Module
  - G_1
  - G_2

Stage 4: Distributed and Parallel Pattern Matching

(a) EDG
- Rule
- E_3
- E_2
- E_1

(b) NFA
- A
- B
- D

Δ = P_1 & P_2 ⇒ True
Δ = P_1 ⇒ True
Δ = P_2 & P_3 ⇒ True

Storage of Archived Streams
- Archived Streams
- Future Streams

Stream_i: Incoming Streams
J_i: Join Operations
E_n: Event Nodes
S_k: Select Operators
G_t: Generated Events
A/B/D: NFA States
PROPOSED MODEL

- Supports Triple based and NG based data model

- Offers event source based Filtering

- Historical management of events through summarisation (Facts Assignments)

- Provide dedicated design for SCEP (No Data or Query Translation unlike EP-SPARQL and other systems)

- Distributed and parallel sub-query processing with query rewriting
**Proposed Model**

- Integrating stream processing and CEP

- Offers various new operators including, Sequencing, Kleene Closure and Negation for RDF Graph patterns

- Allows NFA and EDG to be used in the context of SCEP through query rewriting (from Rule based to State based system)
OVERVIEW

Introduction

SEMANTIC COMPLEX EVENT PROCESSING

PROPOSED APPROACH

CONCLUSION
CONCLUSION

- Annotated RDF NG enables temporal reasoning at RDF level
- Our data/query model and query rewriting allows
  - Annotated NG based event data model
  - Historical management of stream data
  - Integration of various new operators for RDF Graphs (Kleene Closure, Negation)
  - Integration of NFA and EDG in the context of SCEP
  - Parallel and distributed event processing (per query)
Questions?