Towards Unified and Native Enrichment in Event Processing Systems

Souleiman Hasan, Sean O’Riain, Edward Curry

Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland

In proceedings of The 7th ACM International Conference on Distributed Event-Based Systems
June 29 - July 3, 2013, Arlington, Texas, USA

Stefan.Decker@deri.org
http://www.StefanDecker.org/
Talk Overview

- Introduction
  - IoT, Cyber-Physical Systems
  - Event Incompleteness

- Current Approaches

- Proposed Approach
  - Challenges for Enrichment
  - Proposed Model
  - Implications

- Linked Data Instantiation
  - Evaluation

- Summary and Future Directions
Big Data & IoT

Volume in Exabytes

- 2010
- 2015

Percentage of uncertain data

Sensors & Devices
Social Media
VoIP
Enterprise Data

IBM

Enabling networked knowledge.
Cyber-Physical Systems

Smart Grid

Smart City

Smart Building

Smart Enterprise
Event Processing Systems

- Three dimensions of decoupling
  - Removal of explicit dependencies between event producers and consumers
    - Scalable deployment
  - Information exchange only by Events
    - (Eugster et al., 2003)
Problem – Event Incompleteness

- Event producers and consumers are decoupled
  
  Event producers may have very little knowledge about consumers information needs

\[(\text{type}, \text{"energy consumption"})\]  
\[(\text{device}, \text{"heater x"})\]  
\[(\text{consumption}, \text{"high"})\]

\{\text{(type= "energy consumption")} \text{ and } \text{(floor= "first floor")} \text{ and } \text{(consumption=":high")}\}
Dimensions of Incompleteness

- Event Format: lacks syntactical structure
  - E.g. plain text against conjunctive subscription

- Event Semantics: events lack a reference scheme
  - E.g. schema-less tuples

- Lack of Background Knowledge
  - E.g. complementary information exists in external DB

- Incompleteness Addressable by Transformation
  - E.g. transforming amounts of multiple measurement units

- Temporal Segmentation
  - E.g. Complementary information exists in past or future events
Current Approaches
Event Enrichment

IoT Heater Event

(type, "energy consumption")
  (device, "heater x")
  (consumption, "high")

Meta Data

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>ROOM</th>
<th>FLOOR</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>heater x</td>
<td>202e</td>
<td>second floor</td>
<td>white</td>
</tr>
<tr>
<td>heater y</td>
<td>313</td>
<td>third floor</td>
<td>blue</td>
</tr>
</tbody>
</table>

Enriched Event

(type, "energy consumption")
  (device, "heater x")
  (consumption, "high")
  (room, “202e”)
  (floor, “second floor”)
Dedicated agents to complement events
Agent-based Event Enrichment

- **Pros**
  - Events complete with respect to consumer’s need
  - Low false positives/negatives rate

- **Cons**
  - Ad-hoc and external to event processing engines
  - Difficult to develop and maintain enrichment logic
  - Difficult to optimise enrichment process
Proposed Approach
Proposed Approach

- We need
  - Event enrichment to be integrated into the event processing paradigm as a core task of event processing engines

- Proposal
  - Unified declarative language for event processing and enrichment
  - *Enrichment element* as a declarative specification for engine to enrich events with complementary information items
Unified and Native Enrichment
Challenges for Enrichment

1. Determination of Enrichment Source (ES)
   - E.g. BMS relational database via a connection string "Server=www.example.com/rdbms;Database=BMS-DB;".

2. Retrieval of Information Items from the Enrichment Source
   - E.g. SQL query against a query interface

3. Finding the Complementary Information for an Event in the Enrichment Source
   - E.g. recursively join on the ID attribute in the LOCATIONS table and query for $n$ times.
4. Fusion of Complementary Information with the Event

- E.g.

  - (type, "energy consumption")
    - (device, "heater x")
    - (consumption, "high")

  - (type, "energy consumption")
    - (device, "heater x")
    - (consumption, "high")
    - (room, "202e")
    - (floor, "second floor")

- E.g.

  - (type, "energy consumption")
    - (device, "heater x")
    - (consumption, "high")

  - (type, "energy consumption")
    - (device, "heater x")
    - (consumption, "high")
    - (location, "202e, second floor")
Produce or Consumer-side

- **Producer side enrichment**
  - Producers can provide enrichment elements with events
  - Identify enrichment source and mechanism

- **Consumer side enrichment**
  - Have better knowledge of completeness from their perspective
  - Proposed here as a unified element with the subscription matching element
Unified Subscription

Enrichment Element
- ENRICH FROM identify the enrichment source(s)
- RETRIEVE BY specifies retrieval mechanism for atomic information items
- FIND BY dictate retrieval of information items from the enrichment source(s)
- FUSE BY define fusion approach to integrate retrieved data

Matching Element
- as in current matching languages
Formal Model

\[ W = e \cup ES \]

The universe \( U \), the event \( e \), the enrichment source \( ES \), the world \( W \), the enrichment view \( HVS \), and a matching view \( MVS \)
Formal Model

- **Successful Enrichment**
  - Completeness

- **Minimal Successfully Enriched Event**
  - Precision

- **Approximately Minimal Successfully Enriched Event**
  - Cost to turn an enriched event into a minimal successfully enriched event

(More details on formal definition in paper)
Implications

- Sharing & Re-usability of Enrichment Elements
  - Expert consumers can share enrichment knowledge with less informed consumers

- Distribution of Enrichment
  - Distribute enrichment process across nodes to achieve an optimal overall completeness

- Approximation in Event Processing Engines
  - Matching over partially complete events would need to account for still missing information
Linked Data Instantiation
Event Model

Enrichment Source Model

Enrichment source is accessible by dereferencing URIs associated with it as per Principles of Linked Data.
Linked Data Instantiation

- Unified Subscription Language

**ENRICH FROM** http://www.myenterprise.org/devices

**RETRIEVE BY** ‘DEREF’

**FIND BY** ‘Spreading Activation’
‘UniformWeightsAllAdjacent’

**FUSE BY** ‘UNION’

```
{?event rdf:type ont:EnergyConsumption.
?event (?p){3} building:SecondFloor.}
```
Evaluation Dataset

- English Dbpedia (1st of August 2012)
- Constructed from instances of class dbpedia-owl:Event
- Total of 24,000 events
- Each event contains one triple in the form:
  - <eventURI, rdf:type, dbpedia-owl:Event>.
- Enrichment Source: Whole Dbpedia dataset
  - Potnetially 250 million triples for enrichment
Evaluation – FIND BY

- FIND BY - Spreading Activation Strategies
  - **UniformWeightsAllAdjacent**
    - Activation spreads equally to all adjacent nodes
  - **UniformWeightsRandomAdjacent**
    - Activation spreads equally to random set of adjacent nodes
  - **DifferentWeightsSemRel**
    - Activation spreads based on semantic relatedness to terms in matching element of subscription
Path-shaped graph used to generate matching elements

Matching Element

   ?event (?p){1}
   dbpedia:England_national_football_team

   ?event (?p){2}
   dbpedia:Queens_Park_Rangers_F.C.

   ?event (?p){3}
   dbpedia:Loftus_Road.

   ?event (?p){4}
   dbpedia: Fulham_F.C.
Enrichment Results

- SemRel (unified approach) performs best
- Less effective with complex subscriptions
- Completeness and precision *not* weighted equally
Summary and Future Directions
Summary and Future Directions

- Event enrichment can be integrated as a core task of event processing engines
  - Unified enrichment logic with event subscription logic
  - Native enricher tackles incompleteness before matching

- Future Work
  - Formalizing the enrichment language element
  - Leverage commonalities among multiple consumers for enrichment optimization