Event Processing in RDF

Mikko Rinne[1], Eva Blomqvist[2], Robin Keskisärkkä[2], and Esko Nuutila[1]

Workshop on Ontology and Semantic Web Patterns (4th edition) - WOP2013

Short overview for RSP CG 25.9.2013

[1] Department of Computer Science and Engineering, Aalto University, School of Science, Finland firstname.lastname@aalto.fi

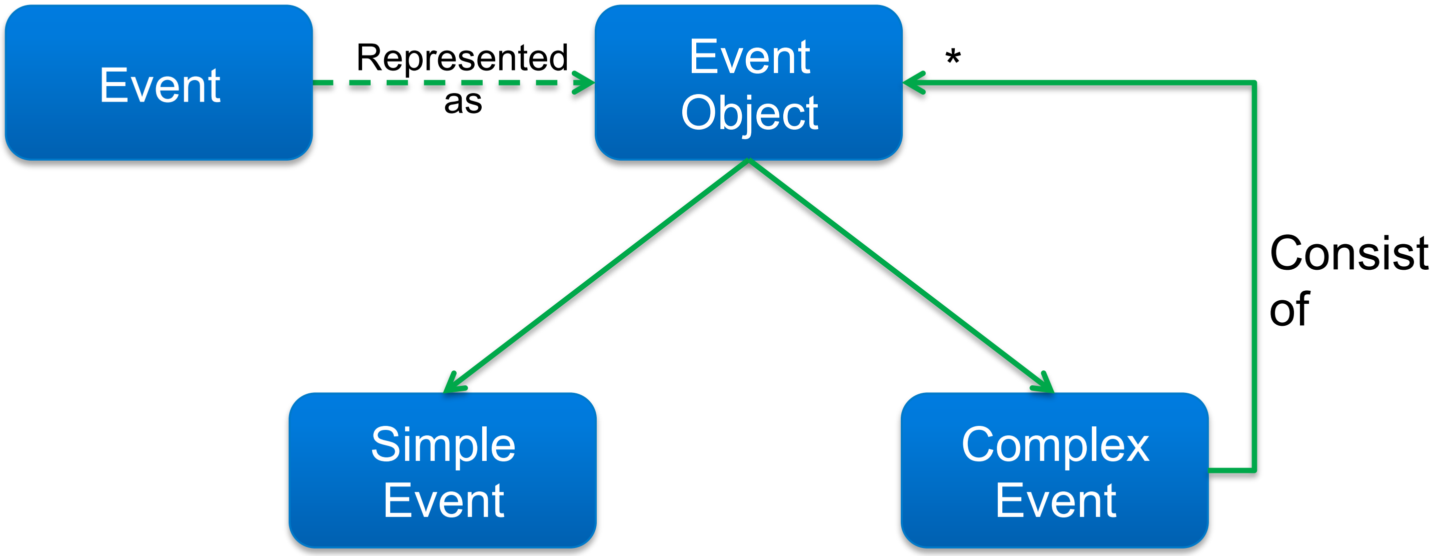
[2] Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden firstname.lastname@liu.se

Sensors and Events in RDF

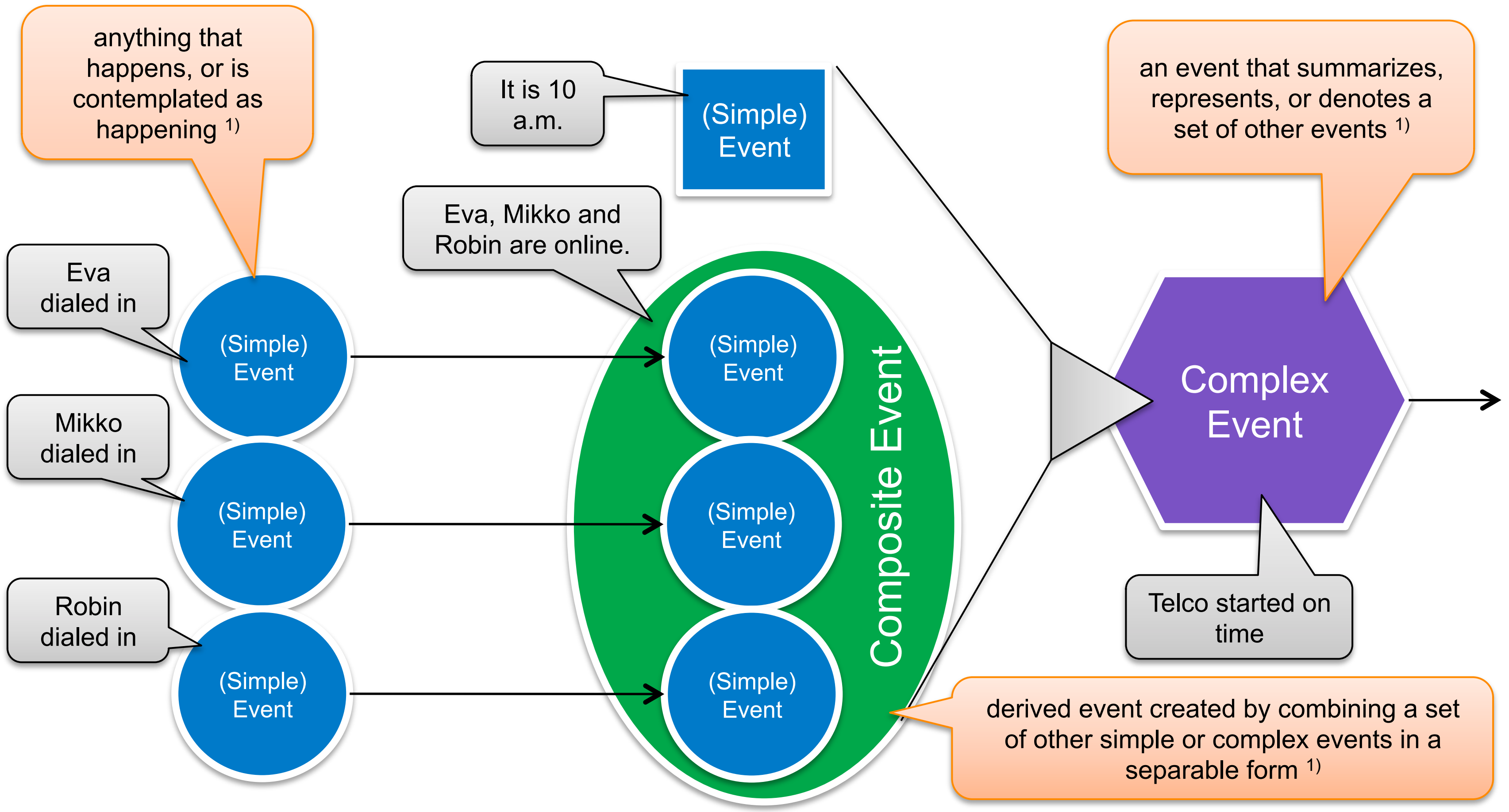
- Multiple ontologies for sensors and events have been defined, e.g.
  - Sensors
    - W3C SSN XG: Semantic Sensor Networks ontology
    - SPITFIRE: Sensor contexts and energy requirements
  - Events
    - Event Ontology: Rooted in music events, but good generality
    - LODE: Linking Open Descriptions of Events
    - Event-F: Comprehensive ontology produced by the WeKnowIt project

Complex Event Processing

- Complex Event Processing pioneered by David Luckham, Opher Etzion, Peter Niblett etc.
- Layered abstractions with self-contained information



Simple, Composite and Complex Events


anything that happens, or is contemplated as happening 1)
It is 10 a.m.
(Simple) Event
an event that summarizes, represents, or denotes a set of other events 1)
Eva, Mikko and Robin are online.
Eva dialed in
(Simple) Event
Mikko dialed in
(Simple) Event
Robin dialed in
(Simple) Event
(Simple) Event
(Simple) Event
(Simple) Event
Composite Event
Complex Event
Telco started on time
derived event created by combining a set of other simple or complex events in a separable form 1)

1) Event Processing Glossary v 2.0

New Requirements for Event Processing Ontology

1. Clear separation of events and event objects
2. Payload support
3. Encapsulated event objects (composite events)
4. References to triggering events
5. Support for multiple timestamps
6. SPARQL querying ability

Proposed Event Processing ODP

- A lightweight re-usable ontology component
- Based on DOLCE Ultra Light
- Aligned with SSN and Event-F, but can be used independently
- Available at: http://ontologydesignpatterns.org/wiki/Submissions:EventProcessing

Composite Event Example

```
:floodWarning0001 a ep:EventObject ;
  ep:hasEventObjectHeader [
    ep:hasEventObjectTime "2013-07-03T08:18:21"^^xsd:dateTime ;
    ep:referstoEventObjectConstituent :weather0001 ;
    ep:refersToEventObjectComponent :waterAlert0001 ;
    …
  ] ; #end of Header
  ep:hasEventObjectBody [
    rdfs:comment "Exemplifies a composite event." ;
  ] . #end of Body

:waterAlert0001 a ep:EventObject ;
  ep:hasEventObjectHeader [
     ep:refersToEventObjectComponent:waterLevel2341 ;
  ] ; #end of Header
  ep:hasEventObjectBody [
    …
  ] .
```

Event Object

Header processed by the event processing system

Reference to triggering event

Reference to component

Payload transported together, but not processed.

Encapsulated event

Encapsulates another event

Querying Events with SPARQL

- Aiming for a generic way to process (copy, move, delete, filter) an event
- Property paths can follow multiple levels of known links
  - Can be used to match encapsulated events in a composite
- Following multiple levels of unknown links (e.g. depth of body) is more risky
  - May accidentally match the whole linked data cloud
  - One possibility is to restrict follow-up to blank nodes (internal to current graph)
  - An explicit OPTIONAL-if combination needed for each level of header or body

Example Query to Match “Floodwarning”

```
CONSTRUCT {
    ?event a ep:EventObject .
    ?event ep:hasEventObjectHeader ?header .
    ?header ?hp ?hv .
    ?header2 ?hp2 ?hv2 .
    ?event ep:hasEventObjectBody ?body .
    ?body ?bp ?bv .
} WHERE {
    :floodWarning0001 ( ep:hasEventObjectHeader /
      ep:refersToEventObjectComponent )* ?event .
    ?event a ep:EventObject .
    ?event ep:hasEventObjectHeader ?header .
    OPTIONAL { ?header ?hp ?hv
      OPTIONAL { BIND ( IF (isBlank(?hv), ?hv, 0) as ?header2)
            ?header2 ?hp2 ?hv2 } }
    OPTIONAL {
      ?event ep:hasEventObjectBody ?body .
      OPTIONAL { ?body ?bp ?bv } }
}
```

Create a copy of the matched composite event object, including encapsulated event objects.

First level headers

Second level nested headers

First level body

Match all encapsulated events

Mandatory header

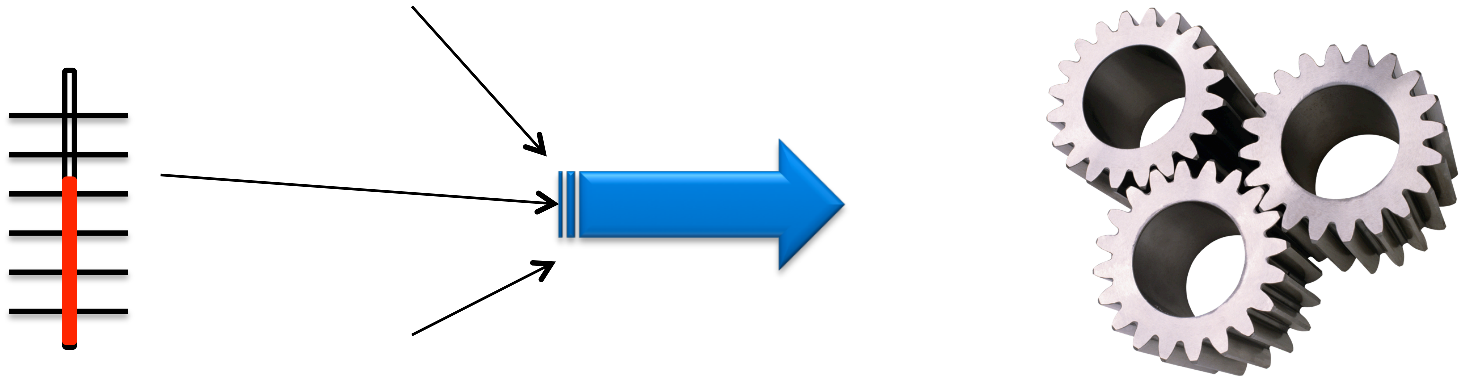
Optional first-level headers

Optional second-level nested headers, only through blank nodes

Optional body

Optional first-level body content

Event Time



hasEventObject-SamplingTime	hasEventObject-ApplicationTime	hasEventObject-SystemTime
Event object sampling time (e.g. recorded by a sensor)	Time of entry to data stream	Time of arrival in the event processing system via the stream

hasEventObjectExpirationTime	any known end time for the event objects validity

Aalto University
School of Science

Summary

- A lightweight ODP to address event processing by RDF and SPARQL has been created and made available
  – All listed requirements have been addressed
- Compatibility with existing solutions
  – All systems based on DUL
  – Especially SSN Ontology and Event-F
- Does not address description and publication of streams
  – E.g. selection of timestamp for windowing should be a stream-specific parameter