RDF Stream Processing for Smart City Applications

Feng Gao, Muhammad Intizar Ali, Alessandra Mileo

INSIGHT Centre for Data Analytics,
National University of Ireland, Galway.
{feng.gao,ali.intizar,alessandra.mileo}@insight-centre.org

Abstract. Processing information flow in real-time is essential in many event-based smart city applications. Such real-time information are often used together with static background knowledge to provide insights on complex situations. We believe RDF stream processing is an ideal tool for addressing both requirements. In this paper we present a demonstration of a workflow for event-based smart city applications that incorporates different RDF stream processing engines.

1 Introduction

RDF Stream Processing (RSP) offers inference capabilities over real-time data flows. A key challenge in applying RSP in large-scale systems, e.g., Smart City applications, is the ability to automatically discovery and integrate heterogeneous data streams. In this demo, we will showcase functionalities of our proposed solution: an event-based middleware called *Automated Complex Event Implementation System (ACEIS)*, which processes application requirements and preferences to automatically discover the most relevant data streams and integrates them to answer users' queries over federated IoT data streams.

2 ACEIS Middleware

Figure 1 shows the architecture of the middle-ware, ACEIS core consists of two main components namely (i) *Resource Management* is responsible for discovering the relevant data streams, and (ii) *Data Federation* generates RDF stream queries following the syntax of underlying engine and dispatches the results of query to the application layer. In the following, we elaborate the key steps in the information flow of ACEIS.

- [Step 1:] Sensor streams are registered as data services within the middleware by providing meta-data information of the stream describing the sensor type, location and sensing capabilities etc. A semantic repository containing sensor description is generated using the Complex Event Service (CES) ontology[1].

[1] CES ontology: http://citypulse.insight-centre.org/ontology/ces

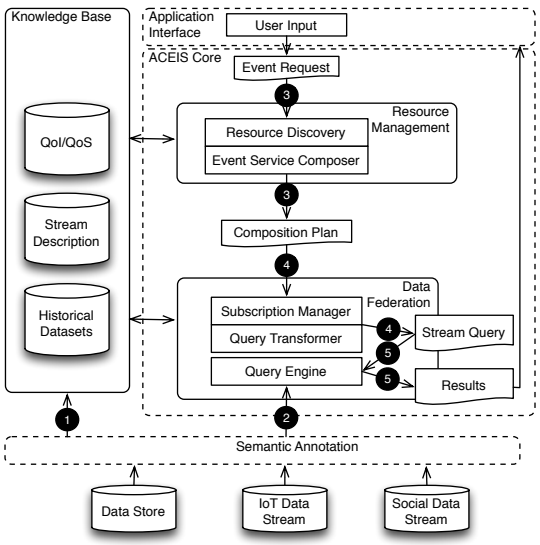

Fig. 1. ACEIS architecture

• [Step 2:] ACEIS consumes semantically annotated data stream produced by the various sensors. We refer our readers to Smart City Information Model for complete description of the Stream Annotation Ontology[2].

• [Step 3:] When an event request is received from the application interface, the resource discovery component utilises static description (including stream quality information annotated with Stream Quality Ontology[3]) of the sensor data streams (as discussed in Step 1) to discover or compose the most relevant data streams for the event request [4, 3]. As a result, a composition plan is generated as a document specifying how relevant streams can be federated to fulfil the event request.

• [Step 4:] The Query Transformer formulates RSP queries from the composition plan generated in previous step. Table 1 indicates the semantic alignment of the event operators with the three state of the art RDF stream processing query languages.

• [Step 5:] formulated queries are executed using RSP query engine and results are continuously produced and dispatched to the application layer.

Demonstration: during the demo session, we will present a travel planner application, which consumes live data streams produced by traffic sensors installed within the City of Aarhus [4] and recommends the best route after considering multi-dimensional factors i.e. traffic congestion level, estimated arrival time, pollution level and users mode of transport etc.

3 Relevance to RSP and ESWC audience:

ACEIS demonstrates how RSP engines can be utilised in real world scenarios to consume live data streams in the context of smart city applications. ACEIS

[2] Stream Annotation Ontology: `http://iot.ee.surrey.ac.uk/citypulse/ontologies/sao/sao`

[3] Stream Quality Ontology: `https://mobcom.ecs.hs-osnabrueck.de/cp_quality/`

[4] Open Data Aarhus: `http://www.odda.dk`

Table 1. Semantics Alignment for Event Operators

ACEIS	E	And	Or	Seq	Rep_o	Rep_n	Sel	Filter	Window
CQELS	SGP	$\bowtie$	$\rtimes\!\!\bowtie$	-	-	-	BGP+$proj$	Filter	Window
C-SPARQL	BGP	$\bowtie$	$\rtimes\!\!\bowtie$	f_t	$\bowtie + f_t$	f_t^+	BGP+$proj$	Filter	Window
ETALIS	BGP	$\bowtie$	$\rtimes\!\!\bowtie$	SeqJoin	$\bowtie$ +SeqJoin	SeqJoin$^+$	BGP+$proj$	Filter	getDuration()

implementation contains both CQELS [5] and C-SPARQL [2] query engine and provides an interesting comparison of the semantics and performance of both engines in real-time use-case scenario. We plan to integrate ETALIS engine [1] as well in future. We believe ACEIS highlights factors that could be helpful for the RSP community's efforts to provide unified query language and semantics for RSP engines. We also believe that ACEIS can attract the attention of the general audience of ESWC by demonstrating RSP in action.

References

1. D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic. Stream reasoning and complex event processing in etalis. *Semant. web*, 3(4):397–407, Oct. 2012.
2. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. C-sparql: Sparql for continuous querying. In *Proceedings of the 18th international conference on World wide web*, pages 1061–1062. ACM, 2009.
3. F. Gao, E. Curry, M. I. Ali, S. Bhiri, and A. Mileo. Qos-aware complex event service composition and optimization using genetic algorithms. In *Service-Oriented Computing*, pages 386–393. Springer, 2014.
4. F. Gao, E. Curry, and S. Bhiri. Complex event service provision and composition based on event pattern matchmaking. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 71–82. ACM, 2014.
5. D. Le-Phuoc, H. Q. Nguyen-Mau, J. X. Parreira, and M. Hauswirth. A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, 16:42–51, 2012.