Sharing and Reusing Continuous Queries - Expression of Interest

Robin Keskisärkkä and Eva Blomqvist

Linköping University, Linköping, Sweden,
firstname.lastname@liu.se

Motivation

The RDF Stream Processing (RSP) community group[1] is in the process of defining a core model for expressing continuous queries over RDF data streams, referred to as the RSP-QL. Previously, a number of RSP query languages have been proposed in conjunction with their corresponding RSP engines, but no consensus has been reached regarding the semantics or syntax of these implementations. For example, C-SPARQL [2], SPARQL$_{stream}$ [3], and CQELS [4] extend standard SPARQL syntax with patterns for defining windows over RDF streams. EP-SPARQL [1] instead extend the SPARQL syntax with focus on temporal operators, supporting an approach more clearly oriented towards event processing. Finally, some engines have used standard compliant SPARQL and SPARQL Update (e.g., INSTANS [5]), relying on users to manually specify, for example, windowing constraints.

Reusing Continuous Queries

In static contexts queries can be validated against predicted output, but developing and validating queries that handle data from several streams of data is less straightforward, and the process can be very time-consuming.

Regardless of the query semantics and syntax of the forthcoming RSP-QL we believe that being able to share and reuse continuous queries will grow in importance. Sharing queries as strings requires them to be adapted purely by hand, making it difficult to share queries efficiently. Representing queries instead as RDF facilitates meta-modeling capabilities, and the possibility of creating query templates.

We propose to extend the SPIN syntax[2] to support the upcoming RSP-QL semantics. The SPIN vocabulary is well-established as a format for representing standard SPARQL 1.1 queries as RDF, and queries can be represented in a syntax-agnostic form. The vocabulary supports meta-modeling of both queries, query templates, and partial queries. Additionally, the official SPIN API supports constraints and inference rules, which could also be leveraged in RSP applications.

[1] https://www.w3.org/community/rsp/
[2] http://topbraid.org/spin/api/

One difficulty in extending SPIN is to verify that the model correctly represents the features of the RSP-QL, which could be discussed at the workshop. We should also consider that the extension can be used to (partially) support engine specific query languages, which could be useful if these engines are maintained.

A possible SPIN syntax extension for representing stream blocks, as supported in CQELS-QL [4], can be seen below. The stream block closely resembles the SPIN representation of a graph block, but with an added blank node for defining the window over the stream.

```
@prefix spin:  <http://spinrdf.org/sp#> .
@prefix rsp: <http://spinrdf.org/sp-ext#> .

[ a rsp:NamedStreamGraph ;
  rsp:streamNameNode <http://www.example/stream> ;
  rsp:window [ a rsp:LogicalWindow ;
               rsp:range [ rsp:value "10" ;
                           rsp:unit "m" ] ;
               rsp:step [ rsp:value "10" ;
                          rsp:unit "s" ] ] ;
  spin:elements ( ... ) ]
```

About the Authors: Robin Keskisärkkä is a PhD student at the Dept. of Computer and Information Science, Linköping University. His research is on event processing and event abstraction using semantic technologies, in particular for the analysis of streaming data in criminal intelligence. Eva Blomqvist is an Assistant Professor in the same department. Her research is focused on decision support systems, in particular for the security and crisis management domain, and is the PhD supervisor of Robin. Both are members of the W3C RSP community group.

References

1. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning. In: Proceedings of the 20th International Conference on World Wide Web (2011)
2. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying RDF streams with C-SPARQL. SIGMOD Record 39(1), 20–26 (2010)
3. Calbimonte, J.P., Corcho, O., Gray, A.J.G.: Enabling Ontology-based Access to Streaming Data Sources. In: Proceedings of the 9th International Semantic Web Conference. pp. 96–111. ISWC'10, Springer-Verlag, Berlin, Heidelberg (2010)
4. Le-Phuoc, D., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In: Proceedings of the 10th International Conference on the Semantic Web. pp. 370–388 (2011)
5. Rinne, M., Nuutila, E., Törmä, S.: INSTANS: High-Performance Event Processing with Standard RDF and SPARQL. In: Proceedings of the ISWC 2012 Posters and Demonstrations Track. Boston, US (2012)