Representing RDF Stream Processing Queries in RSP-SPIN

Robin Keskisärkkä

Linköping University, Linköping, Sweden
robin.keskisarkka@liu.se

Abstract. A number of RDF Stream Processing (RSP) systems have been developed to support processing of streaming Linked Data, however, due to the lack of a standard query language they all provide different extensions. The RSP Community Group[1] is in the process of standardizing the RSP query language (RSP-QL), which incorporates many features from existing RSP languages. In this paper we present a demo showing how RSP-SPIN, a SPIN extension for RSP-QL, can be used to encapsulate RSP-QL queries as RDF, which can then be used to support serialization into multiple RSP languages. This can reduce the effort required to produce and maintain queries for RSP benchmarks, since developers can focus on a single representation per query, and assist developers in combining or switching between different RSP engines.

Keywords: RDF Stream Processing, RSP-QL, RSP-SPIN

1 Introduction

The amount of data published as online streams is increasing. While the streams often contain structured or semi-structured data they are often represented using non-standardized vocabularies and models. Semantic Web (SW) technologies have the potential to make this heterogeneity manageable; however, traditional SW technologies are optimized for performance on more or less static data, and do not scale well when dealing with potentially unbounded streams.

RDF Stream Processing (RSP) has been proposed as a way of bridging the gap between static and streaming Linked Data [11, 10]. Inspired by technologies dealing with similar challenges in other domains several RSP engine implementations have been presented [1–3, 7, 9]. Most of the proposed systems have extended SPARQL to support the definition of sliding windows over RDF streams.

The RSP Community Group[2] is working on defining a standard query language for RSP (RSP-QL), and the current version of the abstract syntax and semantics has already been described in some detail [6, 5]. While the query syntax has yet to be defined in detail the available example queries[3], present a fairly clear picture of what can be expected in a future standard.

[1] https://www.w3.org/community/rsp/
[2] ibid.
[3] See https://github.com/streamreasoning/RSP-QL/

The diversity of RSP implementations, and their corresponding languages, makes performance comparisons and switching between engines complicated. General benchmarks, focusing on different aspects of RSP processing, have been proposed [4, 8, 12], but comparisons also need to based on, for example, volume, velocity, necessary query features, and response-time requirements defined in actual use cases. Developing and validating any type of non-trivial query in a streaming context is usually both difficult and time-consuming. A major difficulty is that it is often not possible to predict the outcome of a query that is run against streaming data. Developers often spend considerable time on generating predicable data streams, or work against recorded data, to be able to compare results with some expected outcome. Additionally, a query expressed for one RSP engine is typically not compatible with another, and some validation step is required for each version of a query.

In this demo we show how RSP-SPIN, an extension of the SPIN Modeling Vocabulary[4], can be used to represent RSP-QL queries in RDF. The demo shows how each query in the CSRBench [4] benchmark can be represented as an RSP-QL query, and we demonstrate serializers from RSP-SPIN to CQELS [7], C-SPARQL [2], and SPARQLstream [3].

2 Architecture and Implementation

Sharing and reusing queries is an important aspect in most contexts where people need to interact with structured data. Many relational databases support some form of stored procedures, which are ready-made parameterized queries that can be instantiated by simply providing the required parameters. Storing queries in such ways has several advantages compared to relying on the users to provide the queries. For example, stored queries dramatically lower the threshold for inexperienced users, can be used to protect against query injections, and to support access control. Stored queries also means that users to not have to have a detailed understanding of the underlying data models, or have any experience in crafting queries themselves.

The SPIN Modeling Vocabulary and the SPIN API was developed in part to provide this functionality for Semantic Web applications. SPIN allows queries to stored as RDF, and enables the representation of parameterized query templates, where query variables can be bound to specific values at runtime.

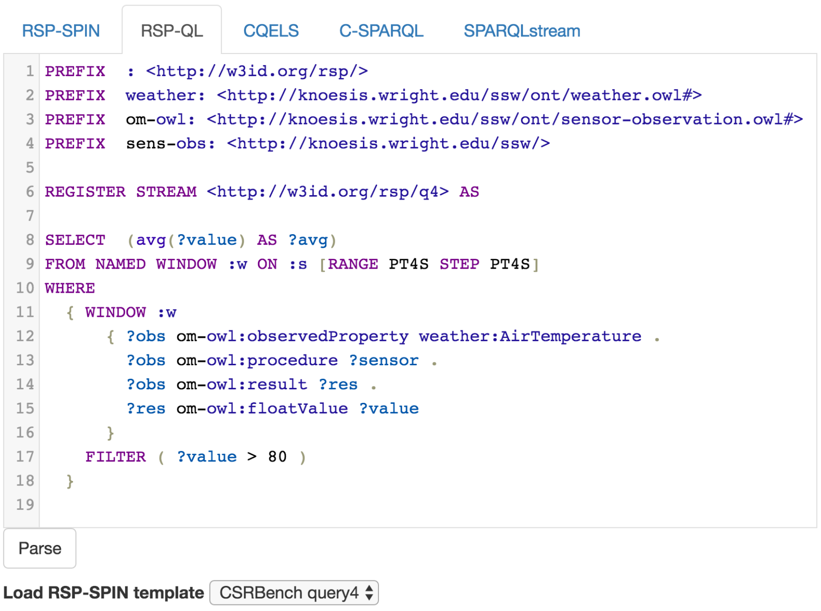
RSP-SPIN extends the functionality of SPIN to model RSP-QL. Since RSP-QL is based on SPARQL the number of changes made to the base model of SPIN is small, and most changes simply involve adding the appropriate properties, classes, and keywords to support the additional concepts introduced in RSP-QL. This means that compatibility with standard SPIN can be maintained.

RSP-SPIN and the RSP-SPIN API are released as open source[5]. The current version has excluded a some features that have been up for discussion in the RSP Group, which have yet to be discussed in sufficient depth. In particular the

[4] https://www.w3.org/Submission/spin-modeling/
[5] https://github.com/keski/rsp-spin/

excluded features include blank nodes for named graphs, and temporal patterns for Complex Event Processing.

The user interface provided for the demo can be seen in Figure 1. The queries are represented as RSP-SPIN, which can be serialized into any of the supported query languages. Activating the RSP-SPIN view displays the underlying RDF model. The user interface does not support user-defined parameters for templates, but this is demonstrated in the predefined queries and on the RSP-SPIN homepage[6].



```
PREFIX  : <http://w3id.org/rsp/>
PREFIX  weather: <http://knoesis.wright.edu/ssw/ont/weather.owl#>
PREFIX  om-owl: <http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#>
PREFIX  sens-obs: <http://knoesis.wright.edu/ssw/>

REGISTER STREAM <http://w3id.org/rsp/q4> AS

SELECT  (avg(?value) AS ?avg)
FROM NAMED WINDOW :w ON :s [RANGE PT4S STEP PT4S]
WHERE
  { WINDOW :w
      { ?obs om-owl:observedProperty weather:AirTemperature .
        ?obs om-owl:procedure ?sensor .
        ?obs om-owl:result ?res .
        ?res om-owl:floatValue ?value
      }
    FILTER ( ?value > 80 )
  }

```

Fig. 1. The user interface with a CSRBench query loaded. The supported formats for RSP-SPIN queries are CQELS, C-SPARLQ, SPARQLstream, and RSP-QL.

Switching between views displays the current query in different formats. The user can pick one of the predefined queries available in the drop-down menu at the bottom, containing the full set of CSRBench queries. The list also includes a few examples that illustrate the expressivity of RSP-QL and how the CQELS, C-SPARQL, and SPARQLstream serializers handle unsupported RSP-QL features. User defined queries can be parsed from any view containing a *parse* button. Pressing the button will pass the text to the web service, which will attempt to create a template for that query. If successful the newly created query will be loaded into the system and all views will be updated accordingly. The live demo is available at **http://ontology.ida.liu.se:8680/iswc/rsp-spin/**.

[6] http://w3id.org/rsp/spin

Acknowledgments This work was partially supported by the EU FP7 project Visual Analytics for Sense-making in Criminal Intelligence Analysis (VALCRI) under grant number FP7-SEC-2013-608142.

References

1. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: A Unified Language for Event Processing and St ream 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 on The Semantic Web – Volume Part I. pp. 96–111. ISWC'10, Springer-Verlag, Berlin, Heidelberg (2010)
4. Dell'Aglio, D., Calbimonte, J.P., Balduini, M., Corcho, O., Della Valle, E.: On Correctness in RDF Stream Processor Benchmarking. In: Proceedings of the 12th International Semantic Web Conference - Part II. pp. 326–342. ISWC '13, Springer-Verlag New York, Inc., New York, NY, USA (2013)
5. Dell'Aglio, D., Calbimonte, J.P., Valle, E.D., Corcho, O.: Towards a Unified Language for RDF Stream Query Processing. In: Gandon, F., Guéret, C., Villata, S., Breslin, J.G., Faron-Zucker, C., Zimmermann, A. (eds.) ESWC (Satellite Events). Lecture Notes in Computer Science, vol. 9341, pp. 353–363. Springer (2015)
6. Dell'Aglio, D., Della Valle, E., Calbimonte, J.P., Corcho, O.: RSP-QL Semantics: A Unifying Query Model to Explain Heterogeneity of RDF Stream Processing Systems. Int. J. Semant. Web Inf. Syst. 10(4), 17–44 (October 2014)
7. 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)
8. Le-Phuoc, D., Dao-Tran, M., Pham, M.D., Boncz, P., Eiter, T., Fink, M.: Linked Stream Data Processing Engines: Facts and Figures. In: Proceedings of the 11th International Conference on The Semantic Web - Volume Part II. pp. 300–312. ISWC'12, Springer-Verlag, Berlin, Heidelberg (2012)
9. 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)
10. Sequeda, J.F., Corcho, O.: Linked Stream Data: A Position Paper. In: Proceedings of the 2nd International Conference on Semantic Sensor Networks. vol. 522, pp. 148–157. CEUR-WS.org, Aachen, Germany, Germany (2009)
11. Valle, E.D., Ceri, S., van Harmelen, F., Fensel, D.: It's a Streaming World! Reasoning upon Rapidly Changing Information . IEEE Intelligent Systems 24(6), 83–89 (2009)
12. Zhang, Y., Duc, P.M., Corcho, O., Calbimonte, J.P.: SRBench: A Streaming RDF/SPARQL Benchmark. In: Proceedings of the 11th International Conference on The Semantic Web. pp. 641–657. ISWC'12, Springer-Verlag, Berlin, Heidelberg (2012)