

# Ontology-based Access to Streaming Data

Jean-Paul Calbimonte<sup>1</sup>, Oscar Corcho<sup>1</sup>, and Alasdair J G Gray<sup>2</sup>

<sup>1</sup>Ontology Engineering Group, Universidad Politécnica de Madrid, Spain  
jp.calbimonte@upm.es, ocorcho@fi.upm.es

<sup>2</sup>School of Computer Science, The University of Manchester, United Kingdom  
a.gray@cs.man.ac.uk

**Abstract.** The availability of streaming data sources is progressively increasing thanks to the deployment of ubiquitous data capturing technologies such as sensor networks. We present an ontology-based streaming data access service, based on extensions to the  $R_2O$  mapping language and its query processor ODEMapster, and to the C-SPARQL RDF stream query language. A preliminary implementation of the approach is also presented. With this proposal we expect to set the basis for future efforts in ontology-based streaming live-data integration.

## 1 Ontology-based Streaming Data Access

In the context of the Semantic Web vision, we aim at providing semantic access to streaming data sources through mappings between the elements in stream and ontological models, as in relational-to-RDF[1] approaches. Our approach consists in creating an Ontology-based Streaming Data Access service (Fig 1).

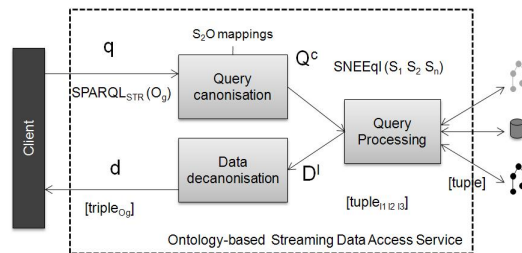


Fig. 1. Ontology-based Streaming Data Access service

The service receives SPARQL<sub>STR</sub> (based on C-SPARQL[2]) queries specified in terms of an ontology. Then the original query is transformed into queries in terms of the sources (*query canonisation*), using a set of  $S_2O$  mappings. These are provided with a-priori knowledge of the ontologies and sources schemas, and are based on the  $R_2O$ [1] mapping language, which has been extended to support streaming queries and data. The transformed query is rewritten in SNEEq[3], a continuous query language that has expressive window and window-to-stream operations, and a semantics that incorporates both streaming and stored data. Afterwards, the query processing phase starts to extract the relevant live-data from the sources and perform the required operations. The result of the query processing will be a set of tuples that will be passed to a *data decanonisation* process, which will transform these tuples to ontology instances.

## 2 Implementation

Our approach has been implemented as an extension to the ODEMapster processor [1]. The implementation generates queries that can be executed by the SNEE streaming query processor using the SNEEql query language [3]. For example in a weather early-warning system, consider a stream `windsamples` and a table `sensors`, and the  $S_2O$  mapping to a `WindSpeedMeasurement` concept (Fig 2):

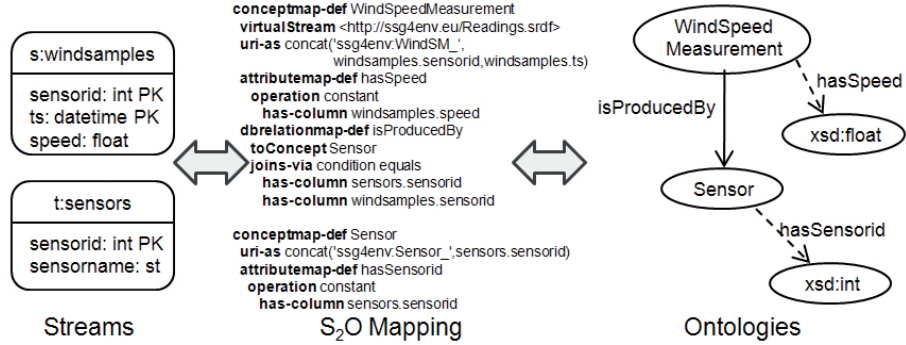


Fig. 2. S<sub>2</sub>O mapping from stream to ontologies

Now we can pose a query over the ontology using SPARQL<sub>STR</sub> for example to obtain the wind speed measurements taken in the last 10 minutes:

```
SELECT ?speed
FROM STREAM <www.ssg4env.eu/SensorReadings.srdf> [RANGE 10 MINUTE STEP 1 MINUTE]
WHERE { ?WindSpeed a fire:WindSpeedMeasurement; fire:hasSpeed ?speed; }
```

The query atoms `WindSpeedMeasurement(x)` and `hasSpeed(x, z)` can be extracted from the SPARQL<sub>STR</sub> query. The window specification is also obtained. As it is defined in the S<sub>2</sub>O mapping the `WindSpeedMeasurement` instances are generated based on the `sensorid` and `ts` attributes of the `windsamples` stream. Similarly the S<sub>2</sub>O mapping defines that `hasSpeed` properties are generated from the values of the speed attribute of the `windsamples` stream. The query generated in the SNEEql language is the following:

```
SELECT RSTREAM concat('http://ssg4env.eu#WindSM', windsensor.id, windsensor.ts )
as id, ( windsamples.speed ) as speed FROM windsamples
[FROM NOW - 10 MINUTE TO NOW SLIDE 1 MINUTE]
```

The results will be transformed into tagged triples, instances of the ontology. As future work, we will investigate incorporating a reasoner on top of the ODEMapster query processor and performing an evaluation of our implementation.

## References

1. Barrasa, J., Corcho, O., Gómez-Pérez, A.: R2O, an extensible and semantically based database-to-ontology mapping language. In: SWDB2004. (2004) 1069–1070
2. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: WWW '09. (2009) 1061–1062
3. Brenninkmeijer, C.Y., Galpin, I., Fernandes, A.A., Paton, N.W.: A semantics for a query language over sensors, streams and relations. In: BNCOD '08. (2008) 87–99