Semantic Sensor Data Search in a Large-Scale Federated Sensor Network

Jean-Paul Calbimonte[1], Hoyoung Jeung[2], Oscar Corcho[1], and Karl Aberer[2]

[1]Ontology Engineering Group, Departamento de Inteligencia Artificial,
Facultad de Informática, Universidad Politécnica de Madrid, Spain
`jp.calbimonte@upm.es,ocorcho@fi.upm.es`
[2]School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
`hoyoung.jeung@epfl.ch,karl.aberer@epfl.ch`

Abstract. Sensor network deployments are a primary source of massive amounts of data about the real world that surrounds us, measuring a wide range of physical properties in real time. However, in large-scale deployments it becomes hard to effectively exploit the data captured by the sensors, since there is no precise information about what devices are available and what properties they measure. Even when metadata is available, users need to know low-level details such as database schemas or names of properties that are specific to a device or platform. Therefore the task of coherently searching, correlating and combining sensor data becomes very challenging. We propose an ontology-based approach, that consists in exposing sensor observations in terms of ontologies enriched with semantic metadata, providing information such as: which sensor recorded what, where, when, and in which conditions. For this, we allow defining virtual semantic streams, whose ontological terms are related to the underlying sensor data schemas through declarative mappings, and can be queried in terms of a high level sensor network ontology.

1 Introduction

Sensors are related to a large number of human activities. They can be found in almost every modern monitoring system, including traffic management, health monitoring, safety services, military applications, environmental monitoring, and location-aware services. In such applications, sensors capture various properties of physical phenomena, hence becoming a major source of streaming data.

This growing use of sensors also increases the difficulty for applications to manage and query sensor data [1]. This difficulty becomes even more noticeable when applications need to search for a particular information set over federated and heterogeneous sensor networks, providing huge volumes of sensor data to large user communities [2]. In these environments, sensors from different vendors and with specific characteristics are installed and added to a system. Each of them produces different values, with different data schemas, precision or accuracy, and in different units of measurement. This heterogeneity complicates the task of querying sensor data as well as the corresponding metadata.

A rich body of research work has addressed the problem of querying data in large-scale sensor networks [3,4,5,6]. These studies generally focused on indexing sensor data, caching query results, and maximizing the shares of data to be carried together over networks. Whilst these methods substantially improve the query processing performance, they do not sufficiently consider the importance and difficulty of heterogeneous (sensor) data integration. In contrast, studies on semantic-aware sensor data management [7,8,9,10,11] have introduced a wide variety of mechanisms that search and reason over semantically enriched sensor data, while considering the heterogeneous characteristics of sensing environments. However, these proposals are still insufficient to show how to manage sensor data and metadata in a federated sensor network, and to efficiently process queries in a distributed environment.

This paper proposes a framework that enables efficient ontology-based querying of sensor data in a federated sensor network, going beyond state-of-the-art storage and querying technologies. The key features of the framework are briefly highlighted as follows:

– Our framework supports semantic-enriched query processing based on ontology information—for example, two users may name two sensors as of types "temperature" and "thermometer", yet the query processing in the framework can recognize that both sensors belong to the same type and include them in query results.
– The framework employs the SSN ontology[1], along with domain-specific ontologies, for effectively modeling the underlying heterogeneous sensor data sources, and establishes mappings between the current sensor data model and the SSN ontology observations using a declarative mapping language.
– The framework enables scalable search over distributed sensor data. Specifically, the query processor first looks up ontology-enabled metadata to effectively find which distributed nodes maintain the sensor data satisfying a given query condition. It then dynamically composes URL API requests to the corresponding data sources at the distributed GSN[2] nodes.
– Our framework has been developed in close collaboration with expert users from environmental science and engineering, and thus reflects central and immediate requirements on the use of federated sensor networks of the affected user community. The resulting system has been running as the backbone of the Swiss Experiment platform[3], a large-scale real federated sensor network.

The paper is organized as follows: we first describe in Section 2 the process of modeling metadata using the SSN ontology, and discuss the mappings between sensor data and the SSN observation model. In Section 3 we introduce the ontology-based query translation approach used in our framework. Section 4 describes the system architecture and its components, and in Section 5 we provide details about technical experimentations of our approach.We then discuss about relevant related work in Section 6, followed by our conclusions in Section 7.

---

[1] W3C Semantic Sensor Network (SSN-XG) Ontology [12]

[2] Global Sensor Networks [13], streaming data middleware used for the prototype.

[3] Swiss-Experiment: `http://www.swiss-experiment.ch/`

2 Modeling Sensor Data with Ontologies

Ontologies provide a formal, usable and extensible model that is suitable for representing information, in our case sensor data, at different levels of abstraction and with rich semantic descriptions that can be used for searching and reasoning [1]. Moreover in a highly heterogeneous setting, using standards and widely adopted vocabularies facilitates the tasks of publishing, searching and sharing the data.

Ontologies have been used successfully to model the knowledge of a vast number of domains, including sensors and observations [14]. Several sensor ontologies have been proposed in the past (see Section 6), some of them focused on sensor descriptions, and others in observations [14]. Most of these proposals are, however, often specific to a project, or discontinued, which do not cover many important areas of the sensor and observation domain. Moreover many of these ontologies did not follow a solid modeling process or did not reuse existing standards. In order to overcome these issues the W3C SSN XG group [12] introduced a generic and domain independent model, the SSN ontology, compatible with the OGC[4] standards at the sensor and observation levels.

The SSN ontology (See Fig. 1) can be viewed and used for capturing various properties of entities in the real world. For instance it can be used to describe sensors, how they function and process the external stimuli. Alternatively it can be centered on the observed data, and its associated metadata [15]. In this study, we employ the latter ontology modeling approach in a large-scale real sensor network application, the Swiss Experiment. For instance consider a wind-monitor sensor in a weather station deployed at a field site. The sensor is capable of measuring the wind speed on its specific location. Suppose that another sensor attached at the same station reports air temperature every 10 minutes. In terms of the SSN ontology both the wind and temperature measurements can be seen as observations, each of them with a different feature of interest (wind and air), and each referring to a different property (speed and temperature).

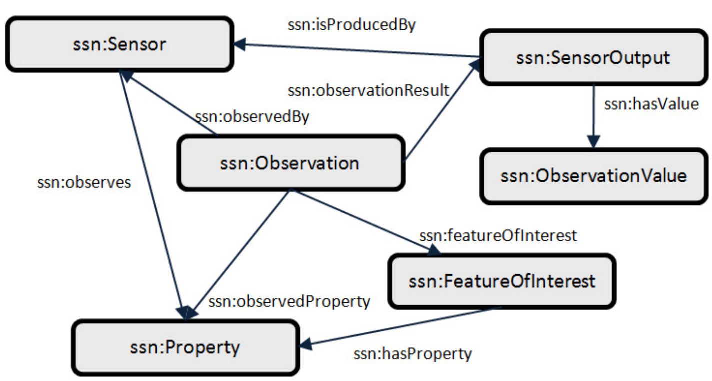

Fig. 1. Main concepts of the SSN ontology.

[4] Open Geospatial Consortium: `http://www.opengeospatial.org/`

In the SSN ontology, instances of the `Observation` class represent such observations, e.g. Listing 1.1, and are linked to a certain feature instance through a `featureOfInterest` property. Similarly the `observedProperty` links to an instance of a property, such as speed. Since the SSN model is intended to be generic, it does not define the possible types of observed properties, but these can be taken from a specialized vocabulary such as the NASA SWEET[5] ontology. Actual values of the sensor output can also be represented as instances linked to the `SensorOutput` class through the `hasValue` property. The data itself can be linked through a specialized property of a quantity ontology (e.g. the QUDT[6] `numericValue` property). Finally the observation can be linked to a particular sensor (e.g. Sensor instance `SensorWind1` through the `observedBy` property). Evidently more information about the observation can be recored, including units, accuracy, noise, failures, etc. Notice that the process of ontology modeling requires reuse and combination of the SSN ontology and domain-specific ontologies.

```
swissex:WindSpeedObservation1 rdf:type ssn:Observation;
      ssn:featureOfInterest [ rdf:type sweet:Wind];
      ssn:observedProperty  [ rdf:type sweetProp:Speed].
      ssn:observationResult
         [ rdf:type ssn:SensorOutput;
           ssn:hasValue [qudt:numericValue "6.245"^^xsd:double]];
      ssn:observedBy swissex:SensorWind1;
```

Listing 1.1. Wind Speed observation in RDF according to the SSN ontology

In our framework, we also model the sensor metadata. For example we can specify that the weather station platform where both sensors are installed, is geospatially located, using the SG84 vocabulary[7]. In the example in Listing 1.2, the location (latitude and longitude) of the platform of the `SensorWind1` sensor is provided. We can also include other information such as a responsible person, initial date of the deployment, etc.

```
swissex:SensorWind1 rdf:type ssn:Sensor;
   ssn:onPlatform [:hasGeometry [rdf:type wgs84:Point;
                                 wgs84:lat "46.8037166";
                                 wgs84:long "9.7780305"]];
   ssn:observes [rdf:type sweetProp:WindSpeed] .
```

Listing 1.2. Representation of a Sensor on a platform and its location in RDF

Although the observation model provides a semantically enriched representation of the data, sensors generally produce streams of raw data with very little structure and thus there is a gap between the observation model and the original data. For instance both sensors in Listing 1.3 (`wan7` and `imis_wfbe`) capture wind speed measurements but have different schemas, each one stores the observed value in a different attribute. To query wind speed observations in these

[5] `http://sweet.jpl.nasa.gov/` NASA SWEET Ontology
[6] Quantities, Units, Dimensions and Data Types ontologies, `http://www.qudt.org/`
[7] Basic Geo WGS84 Votcabulary: `http://www.w3.org/2003/01/geo/`

settings, the user needs to know the names of the sensors, and the names of all different attributes that match with the semantic concept of wind speed. This is an error-prone task and is unfeasible when the number of sensors is large.

```
wan7: {wind_speed_scalar_av FLOAT, timed DATETIME}
imis_wbfe: {vw FLOAT, timed DATETIME}
```

Listing 1.3. Heterogeneous sensor schemas

We take an ontology mapping-based approach to overcome this problem. Although in previous works [16,17] sensor observations are provided and published as RDF and linked data, they do not provide the means and representation that allows querying live sensor data in terms of an ontological model. Going beyond these approaches, we propose using declarative mappings that express how to construct SSN Observations from raw sensor schemas, and for this purpose we use the W3C RDB2RDF Group, R2RML language[8] to represent the mappings. For example we can specify that for every tuple of the `wan7` sensor, an instance of a SSN `ObservationValue` must be created, using the mapping definition `Wan7WindMap` depicted in Fig. 2 (See Listing 1.4 for its R2RML representation).

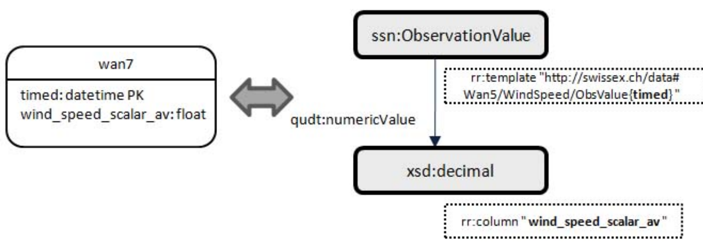

Fig. 2. Simple mapping from the `wan7` sensor to a SSN `ObservationValue`

The instance URI is composed according to the mapping `rr:template` rule that concatenates the `timed` column value to a prefix. The observation actual value is extracted from the `wind_speed_scalar_av` sensor field and is linked to the `ObservationValue` through a `qudt:numericValue` property.

```
:Wan7WindMap a rr:TriplesMapClass;
   rr:tableName "wan7";
   rr:subjectMap
     [rr:template
        "http://swissex.ch/data#Wan5/WindSpeed/ObsValue{timed}";
      rr:column "timed";
      rr:class ssn:ObservationValue;
      rr:graph swissex:WannengratWindSpeed.srdf ];
   rr:predicateObjectMap
     [ rr:predicateMap [ rr:predicate qudt:numericValue ];
       rr:objectMap    [ rr:column "wind_speed_scalar_av" ] ]; .
```

Listing 1.4. Mapping a sensor to a SSN ObservationValue in R2RML

[8] R2RML mapping language, `http://www.w3.org/2001/sw/rdb2rdf/r2rml/`

By using the mappings and the SSN ontology, we are able to express the sensor metadata and observations data using a semantic model, even if the underlying data sources are relational streams. In the next section we provide details about the query translation process that is carried out to make querying possible.

3 Querying Ontology-based Sensor Data

Ontology-based streaming data access aims at generating semantic web content from existing streaming data sources [18]. Although previous efforts have been made in order to provide semantic content automatically form relational databases using mappings [19], only recently this idea has been explored in the context of data stream management [18]. Our approach in this paper (Fig. 3) covers this gap, extending the work of [18] to support the R2RML syntax and produce algebra expressions that can be transformed into requests to federated sensor networks.

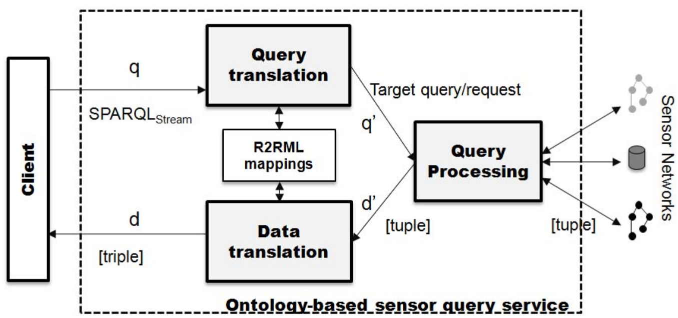

Fig. 3. Ontology-based sensor query service: translation of $\text{SPARQL}_{\text{Stream}}$ queries over virtual RDF streams, to requests over federated sensor networks

Our ontology-based sensor query service receives queries specified in terms of the SSN ontology using $\text{SPARQL}_{\text{Stream}}$ [18], an extension of SPARQL that supports operators over RDF streams such as time windows, and has been inspired by C-SPARQL [8]. Since the $\text{SPARQL}_{\text{Stream}}$ query is expressed in terms of the ontology, it has to be transformed into queries in terms of the data sources, using a set of mappings, expressed in R2RML. The language is used to define declarative mappings from relational sources to datasets in RDF, as detailed in Section 2. These are in fact *virtual* RDF *streams*, since they are not materialized beforehand, but the data is queried and transformed on demand after the $\text{SPARQL}_{\text{Stream}}$ query is translated. The target of this *query translation* process is a streaming query expression over the sensor streams. These queries are represented as algebra expressions extended with time window constructs, so that optimizations can be performed over them and can be easily translated to a target language or stream request, such as an API URL, as we will see in Section 4.

As an example, consider the mapping in Fig. 4, which extends the one displayed before in Fig. 2. This mapping generates not only the `ObservationValue` instance but also a `SensorOutput` and an `Observation` for each record of the sensor `wan7`. Notice that each of these instances constructs its URI with a different template rule and the `Observation` has a `observedProperty` property to the `WindSpeed` property defined in the SWEET ontology.

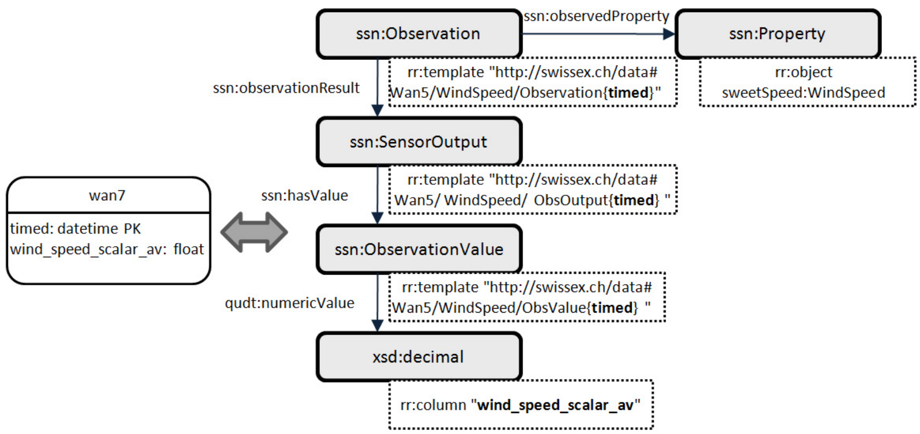

Fig. 4. Mapping from the `wan7` sensor to a `Observation` and its properties

The following query (Listing 1.5), obtains all wind-speed observation values greater than some threshold (e.g. 10) in the last 5 hours, from the sensors virtual RDF stream `swissex:WannengratWindSensors.srdf`. Such queries are issued by geo-scientists to collect filtered observations and feed their prediction models.

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX swissex: <http://swiss-experiment.ch/metadata#>
PREFIX qudt: <http://data.nasa.gov/qudt/owl/qudt#>
PREFIX sweetSpeed: <http://sweet.jpl.nasa.gov/2.1/propSpeed.owl#>
SELECT ?speed ?obs
FROM NAMED STREAM swissex:WannengratWindSpeed.srdf [NOW - 5 HOUR ]
WHERE {
  ?obs       a ssn:Observation;
             ssn:observationResult ?result;
             ssn:observedProperty ?prop.
  ?prop      a sweetSpeed:WindSpeed.
  ?result    ssn:hasValue ?obsvalue.
  ?obsvalue  a ssn:ObservationValue;
             qudt:numericValue ?speed.
  FILTER ( ?speed > 10 )  }
```

Listing 1.5. SPARQL$_{\text{Stream}}$ query

Using the mapping definitions, the query translator can compose the corresponding algebra expression that creates a time window of 5 hours over the `wan7` sensor, applies a selection with the predicate `wind_speed_scalar_av > 10`, and finally projects the `wind_speed_scalar_av` and `timed` columns (See Fig. 5).

The algebra expressions can be transformed to continuous queries in languages such as CQL [20] or SNEEql [21], and then executed by a streaming query engine. In the case of GSN as the query engine, the algebra expression can be used to produce a sensor data request to the stream query engine. Specifically,

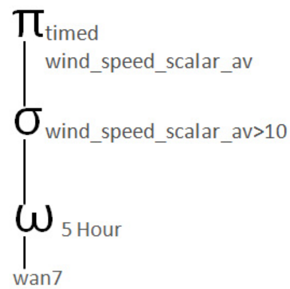

Fig. 5. Translation of the query in Listing 1.5 to an algebra expression, using the R2RML mappings.

the query engine in our framework processes the requests and returns a result set that matches the SPARQL$_{\text{Stream}}$ criteria. To complete the query processing, the result set is transformed by the *data translation* process to ontology instances (SPARQL bound variables or RDF, depending if it is a SELECT or a CONSTRUCT query).

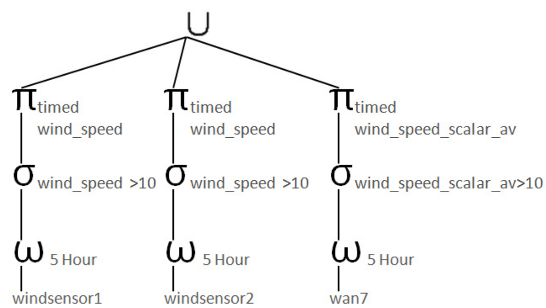

Fig. 6. Algebra UNION expression, with two additional wind-speed sensors.

Depending on the mappings available, the resulting algebra expression can become entirely different. For instance, suppose that there are similar mappings for the `windsensor1` and `windsensor2` sensors, also measuring wind-speed values as `wan7`. Then the resulting expression would be similar to the one in Fig. 6, but including all three sensors in a UNION expression. Conversely, a mapping for a sensor that observes a property different than `sweetSpeed:WindSpeed` will be ignored in the translation process for the sample query.

4 System Overview

Using the ontology-based approach for streaming data described in the previous section, we have built a sensor data search prototype implementation for the Swiss-Experiment project. The system (Fig. 7) consists of the following main components: the user interface, the federated GSN stream server instances, the sensor metadata repository and the ontology-based sensor query processor.

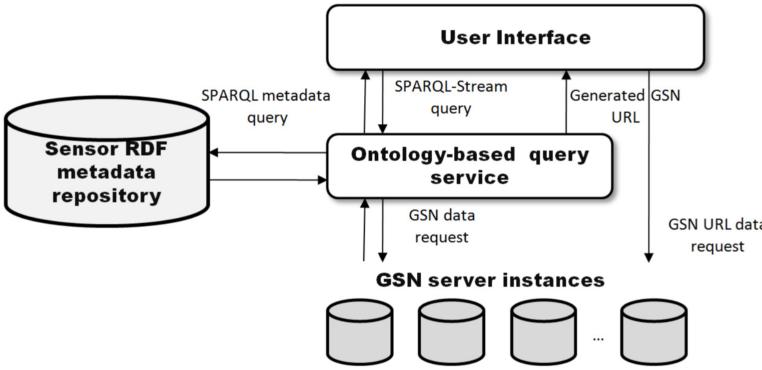

Fig. 7. System architecture

4.1 User Interface

The web-based user interface is designed to help the user filtering criteria to narrow the number of sensors to be queried (Fig. 8). Filtering criteria may include the sensing capabilities of the devices, e.g. select only the sensors that measure air temperature or wind speed. It is also possible to filter according to the characteristics of the deployment or platform, e.g. select sensors deployed in a particular region, delimited by a geo-referenced bounding box. It is also possible to filter by both data and metadata parameters. For instance the user may filter only those sensors registering air temperature values higher than 30 degrees. The filtering parameters can be passed to the ontology-based query processor, as a SPARQL$_{\text{Stream}}$ query in terms of the SSN ontology as detailed next.

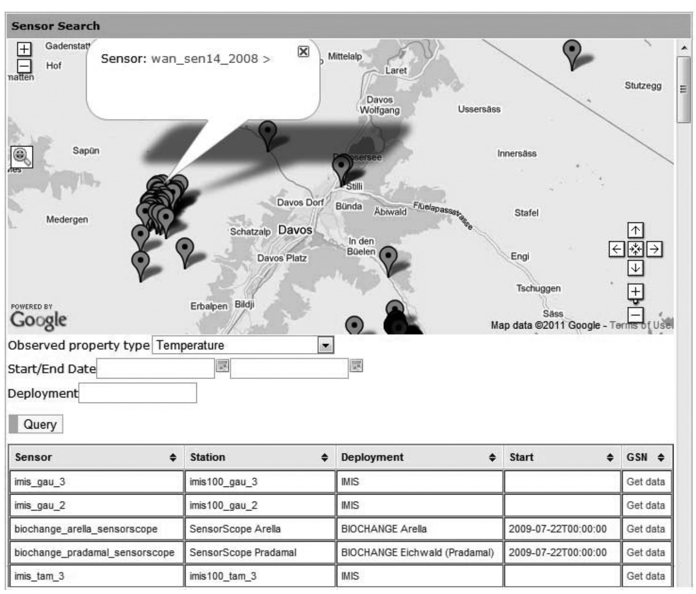

Fig. 8. Sensor data search user interface

4.2 Ontology-based Sensor Query Processor

This component is capable of processing the SPARQL$_{\text{Stream}}$ queries received from the user interface, and perform the query processing over the metadata repository and the GSN stream data engine. The ontology-based processor uses the previously defined R2RML mappings and the sensor metadata in the RDF repository to generate the corresponding requests for GSN, as explained in Section 3.

The ontology-based query service delegates the processing to the GSN server instances by composing *data requests* according to the GSN web-service or URL interfaces. In the case of the web service, a special GSN wrapper for the WSDL specification[9] has been developed, that can be used if the user requires to obtain the observations as RDF instances, just as described in Section 3. Alternatively, the ontology-based sensor query processor can generate GSN API[10] URLs from the algebra expressions. These URLs link directly to the GSN server that provides the data with options such as bulk download, CSV formatting, etc.

```
http://montblanc.slf.ch:22001/multidata?vs[0]=wan7&
      field[0]=wind_speed_scalar_av&
      from=15/05/2011+05:00:00&to=15/05/2011+10:00:00&
      c_vs[0]=wan7s&c_field[0]=wind_speed_scalar_av&c_min[0]=10
```

Listing 1.6. Generation of a GSN API URL

For example, the expression in Fig. 5 produces the GSN API URL in Listing 1.6. The first part is the GSN host (`http://montblanc.slf.ch:22001`). Then the sensor name and fields are specified with the `vs` and `field` parameters. The `from-to` part represents the time window and finally the last line specifies the selection of values greater than 10 (with the `c_min` parameter). These URLs are presented in each sensor info-box in the user interface map.

With this semantically enabled sensor data infrastructure, users can issue complex queries that exploit the existing relationships of the metadata and also the mappings, such as the one in (Listing 1.7).

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX omgeo: <http://www.ontotext.com/owlim/geo#>
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
PREFIX swissex: <http://swiss-experiment.ch/metadata#>
PREFIX sweet: <http://sweet.jpl.nasa.gov/2.1/prop.owl#>
SELECT ?obs ?sensor
FROM NAMED STREAM swissex:WannengratSensors.srdf [NOW - 5 HOUR ]
WHERE {
  ?obs        a ssn:Observation;
              ssn:observedBy ?sensor.
  ?sensor     ssn:observes ?prop;
              ssn:onPlatform ?platform.
  ?platform   dul:hasLocation [swissex:hasGeometry ?geo].
  ?geo        omgeo:within(46.85 9.75 47.31 10.08) .
  ?prop       a sweet:MotionProperty.      }
```

Listing 1.7. SPARQL$_{\text{Stream}}$ query for the ontology-based sensor metadata search

[9] GSN Web Service Interface: `http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/misc/gsn-webservice-api.pdf`

[10] GSN Web URL API: `http://sourceforge.net/apps/trac/gsn/wiki/web-interfacev1-server`

This query requests the observations and originating sensor in the last 5 hours, for the region specified by a bounding box, and only for those sensors that measure motion properties. The geo-location query boundaries are specified using the `omgeo:within` function, and RDF semantic stores such as OWLIM [11] use semantic spatial indexes to compute these kind of queries. Regarding the observed property, considering that the `MotionProperty` is defined in the SWEET ontology as a superclass of all motion-related properties such as Wind Speed, Acceleration or Velocity, all sensors that capture these properties are considered in the query.

In all these examples, the users do not need to know the particular names of the real sensors, nor they need to know all the sensor attribute names that represent an observable property. This clearly eases the task for a research scientist, who can easily use and access the data he needs, with little knowledge of the technical details of the heterogeneous sensor schemas and their definitions. Also, this framework enables easily plugging new sensors to the system, without changing any existing query and without programming. All previous queries would seamlessly include new sensors, if their metadata and mappings are present in the repository.

4.3 GSN Server Instances

Our ontology-based approach for sensor querying relies on the existence of efficient stream query engines that support live sensor querying and that can be deployed in a federated environment. In the Swiss-Experiment project, the sensor data is maintained with Global Sensor Networks (GSN)[13], a processor that supports flexible integration of sensor networks and sensor data, provides distributed querying and filtering, as well as dynamic adaptation and configuration.

The Swiss-Experiment project has several GSN instances deployed in different locations which operate independently. In this way they can efficiently perform their query operations locally, and can be accessed using the interfaces mentioned earlier. However the metadata for these instances is centralized in the RDF metadata repository, enabling the federation of these GSN instances as described in the previous subsection.

4.4 Sensor Metadata Repository

We have used the Sesame [12] RDF store for managing the centralized sensor metadata, using the SSN ontology.The entire set of sensor metadata is managed with the Sensor Metadata Repository (SMR)[2]. The SMR is a web-based collaborative environment based on Semantic Wiki technologies [22], which includes not only static metadata but also dynamic metadata including the information of outliers and anomalies or remarks on particular value sets. This system provides

[11] OWLIM: `http://www.ontotext.com/owlim`
[12] Sesame: `http://www.openrdf.org/`

an easy and intuitive way of submitting and editing their metadata without any programming.

In SMR each sensor, platform or deployment has an associated Wiki page where the data can be semantically annotated with attribute-value pairs, and entities can be connected to each other with semantic properties. This allows interlinking related pages and also dynamically generating rich content for the users, based on the annotated metadata. The entire contents of the SMR can be queried programmatically using the SPARQL language, making it usable not only for humans but also for machines.

5 Experimentation

In order to validate our approach we have conducted a series of experiments in the sensor data and metadata system described previously. The goals were to (i) analyze empirically the scalability of semantic sensor metadata queries and (ii) assess the query and data transformation overhead of our approach. For the first objective, we compared a straightforward (but currently used by scientists) way of obtaining all sensors that measure a particular property (e.g. temperature), with our approach. The former consists in getting sensor details form every sensor in every deployment in the distributed system, and then comparing the sensor attribute name with the property name.

In our environment we have 28 deployments (aprox. 50 sensors in each one), running on its own GSN instance accessible through a web service interface. Therefore to perform this operation the client must contact all of these services to get the required information, making it very inefficient as the number of deployments increases (See Fig. 9). Conversely, using our centralized semantic search we eliminated the need of contacting the GSN instances at all for this type of query, as it can be solved by exploring the sensor metadata, looking for those sensors that have a `ssn:observes` relationship with the desired property.

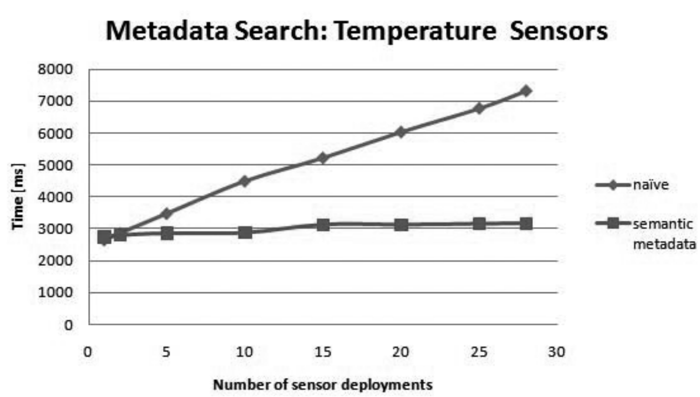

Fig. 9. Comparing metadata search: obtain all sensors that measure temperature. The naïve vs. semantic centralized approach.

As we see in Fig. 9 it is not only scalable as we add more deployments, but we also provide an answer that is independent of the syntactic name assigned to the sensor attributes.

Our approach sometimes incurs in a computing overhead when translating the SPARQL$_{\text{Stream}}$ queries to the internal algebra and the target language or URL request, using the mapping definitions. We analyzed this by comparing the query times of a raw GSN service request and a SPARQL$_{\text{Stream}}$ query translated to an equivalent GSN request. We executed this test over a single simulated deployment, first with only one sensor and up to 9 sensors with data updates every 500 ms. The query continuously obtains observations from the sensors in the last 10 minutes, filtering values smaller than a fixed constant, similarly to Listing 1.5.

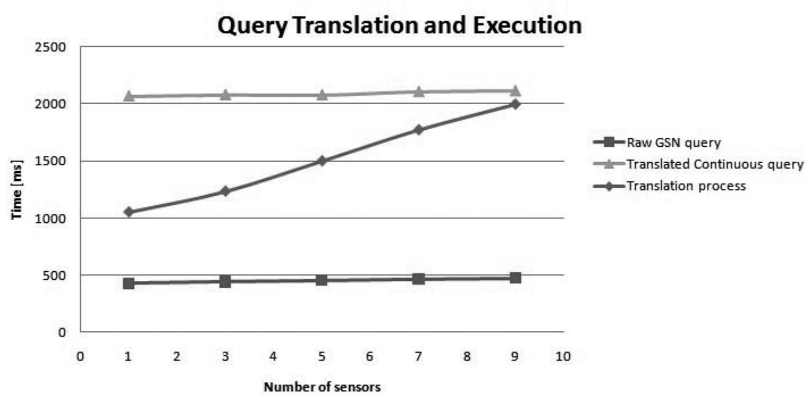

Fig. 10. Query execution and translation overhead: comparing a raw query vs. query translation.

As we show in Fig. 10 the overhead is of roughly 1.5 seconds for the test case. Notice that the overhead is seemingly constant as we add more sensors to the mappings. However this is a continuous query and the translation time penalty has been excluded form the computation, as this operation is only executed once, then the query can be periodically executed. In any case this additional overhead is also displayed in Fig. 10 and it degrades as the number of mappings to sensors increases. This is likely because mappings are stored and loaded as files, and not cached in any way. More efficient management of large collections of mappings could throw better results for the translation operation. Nevertheless we show that continuous queries have an acceptable overhead, almost constant for the chosen use-case.

6 Related Work

Several efforts in the past have addressed the task of representing sensor data and metadata using ontologies, and also providing semantic annotations and querying over these sources, as recounted below.

Ontology Modeling for Sensor Data The task of modeling sensor data and metadata with ontologies has been addressed by the semantic web research community in recent years. As recounted in [14], many of the early approaches focused only on sensor meta-information, overlooking observation descriptions, and also lacked the best practices of ontology reuse and alignment with standards. Recently, through the W3C SSN-XG group, the semantic web and sensor network communities have made an effort to provide a domain independent ontology, generic enough to adapt to different use-cases, and compatible with the OGC standards at the sensor level (SensorML[13]) and observation level (O&M[14]). These ontologies have also been used to define and specify complex events and actions that run on an event processing engine [23].

Semantic Sensor Queries and Annotations Approaches providing search and query frameworks that leverage semantic annotations and metadata, have been presented in several past works. The architectures described in [24] and [25], rely on bulk-import operations that transform the sensor data into an RDF representation that can be queried using SPARQL in memory, lacking scalability and the real-time querying capabilities.

In [10] the authors describe preliminary work about annotating sensor data with Linked Data, using rules to deduce new knowledge, although no details about the RDF transformation are provided. Semantic annotations are also considered for the specific task of adding new sensors to observation services in [9]. The paper points out the challenges of dynamically registering sensors, including grounding features to defined entities, to temporal, spatial context. In [2], the authors describe a metadata management framework based on Semantic Wiki technology to store distributed sensor metadata. The metadata is available through SPARQL to external services, including the system's sensor data engine GSN, that uses this interface to compute distributed joins of data and metadata on its queries.

In [26] a semantic annotation and integration architecture for OGC-compliant sensor services is presented. The approach follows the OGC-sensor Web enablement initiative, and exploits semantic discovery of sensor services using annotations. In [11] a SOS service with semantic annotations on sensor data is defined. The approach consists in adding annotations, i.e. embed terminology form an ontology in the XML O&M and SensorML documents of OGC SWE, using either XLink or the SWE swe:definition attribute for that purpose. In a different approach, the framework presented in [27] provides sensor data readings annotated with metadata from the Linked Data Cloud. While in this work we addressed the

[13] OGC SensorML: `http://www.opengeospatial.org/standards/sensorml`
[14] Observations & Measurements: `http://www.opengeospatial.org/standards/om`

problems related to heterogeneity of the data schemas, it is also worth mentioning that Linked Data initiatives can be helpful for integrating data from different (local or remote) publishers, unlike our use case where all the observations were centralized through GSN.

7 Conclusions

We presented an ontology-based framework for querying sensor data, considering metadata and mappings to underlying data sources, in a federated sensor network environment. Our approach reuses the SSN ontology along with domain-specific ontologies for modeling the sensor metadata so that users can pose queries that exploit their semantic relationships, therefore they do not require any knowledge about sensor specific names or their attributes or schemas. Users can just issue a high-level query that will internally look for the appropriate and corresponding sensors and attributes, according to the query criteria.

For this purpose we perform a dynamic translation of SPARQL$_{\text{Stream}}$ queries into algebra expressions that can be used to generate queries or data requests like the GSN API URLs, while extending the use of the R2RML language specification for streaming sensor data. As a result we have enabled distributed processing of queries in a federated sensor network environment, through a centralized semantic sensor metadata processing service. This approach has been implemented in the Swiss-Experiment project, in collaboration with users form the environmental science community, and we have built a sensor search prototype powered by our framework. We are planning to expand this work in the future, to integrate this platform with external data sources that may provide additional information about the sensors, including location, features of interest or other metadata. Finally we are considering the integration with other sensor data sources running under other platforms, which may be relevant in the domain.

Acknowledgements Supported by the myBigData project (TIN2010-17060) funded by MICINN (Spanish Ministry of Science and Innovation), and the european projects PlanetData (FP7-257641) and SemSorGrid4Env (FP7-223913).

References

1. Corcho, O., García-Castro, R.: Five challenges for the Semantic Sensor Web. Semantic Web **1**(1) (2010) 121–125
2. Jeung, H., Sarni, S., Paparrizos, I., Sathe, S., Aberer, K., Dawes, N., Papaioannou, T., Lehning, M.: Effective Metadata Management in Federated Sensor Networks. In: SUTC, IEEE (2010) 107–114
3. Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G., Olston, C., Rosenstein, J., Varma, R.: Query processing, resource management, and approximation in a data stream management system. In: CIDR. (2003) 245–256
4. Ahmad, Y., Nath, S.: COLR-Tree: Communication-efficient spatio-temporal indexing for a sensor data web portal. In: ICDE. (2008) 784–793

5. Li, J., Deshpande, A., Khuller, S.: Minimizing communication cost in distributed multi-query processing. In: ICDE. (2009) 772 –783
6. Wu, J., Zhou, Y., Aberer, K., Tan, K.L.: Towards integrated and efficient scientific sensor data processing: a database approach. In: EDBT. (2009) 922–933
7. Compton, M., Neuhaus, H., Taylor, K., Tran, K.: Reasoning about sensors and compositions. In: SSN. (2009)
8. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: WWW '09, ACM (2009) 1061–1062
9. Bröring, A., Janowicz, K., Stasch, C., Kuhn, W.: Semantic challenges for sensor plug and play. Web and Wireless Geographical Information Systems (2009) 72–86
10. Wei, W., Barnaghi, P.: Semantic annotation and reasoning for sensor data. In: Smart Sensing and Context. (2009) 66–76
11. Henson, C., Pschorr, J., Sheth, A., Thirunarayan, K.: SemSOS: Semantic Sensor Observation Service. In: CTS, IEEE Computer Society (2009) 44–53
12. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K.: Semantic Sensor Network XG final report, available at `http://www.w3.org/2005/Incubator/ssn/XGR-ssn/`. Technical report, W3C Incubator Group (2011)
13. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: VLDB, VLDB Endowment (2006) 1199–1202
14. Compton, M., Henson, C., Lefort, L., Neuhaus, H., Sheth, A.: A survey of the semantic specification of sensors. In: SSN. (2009) 17
15. Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In: SSN. (2010) 7–11
16. Patni, H., Henson, C., Sheth, A.: Linked sensor data. In: Collaborative Technologies and Systems (CTS), 2010 International Symposium on, IEEE (2010) 362–370
17. Barnaghi, P., Presser, M., Moessner, K.: Publishing Linked Sensor Data. In: SSN. (2010)
18. Calbimonte, J., Corcho, O., Gray, A.: Enabling ontology-based access to streaming data sources. In: ISWC. (2010) 96–111
19. Sahoo, S.S., Halb, W., Hellmann, S., Idehen, K., Jr, T.T., Auer, S., Sequeda, J., Ezzat, A.: A survey of current approaches for mapping of relational databases to RDF. W3C (January 2009)
20. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. The VLDB Journal **15**(2) (June 2006) 121–142
21. 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
22. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic Wikipedia. In: WWW '06, ACM (2006) 585–594
23. Taylor, K., Leidinger, L.: Ontology-driven complex event processing in heterogeneous sensor networks. In: ESWC. (2011) 285–299
24. Lewis, M., Cameron, D., Xie, S., Arpinar, B.: ES3N: A semantic approach to data management in sensor networks. In: SSN. (2006)
25. Huang, V., Javed, M.: Semantic sensor information description and processing. In: SENSORCOMM, IEEE (2008) 456–461
26. Babitski, G., Bergweiler, S., Hoffmann, J., Schön, D., Stasch, C., Walkowski, A.: Ontology-based integration of sensor web services in disaster management. GeoSpatial Semantics (2009) 103–121
27. Le-Phuoc, D., Parreira, J., Hausenblas, M., Han, Y., Hauswirth, M.: Live linked open sensor database. In: I-Semantics, ACM (2010) 1–4