Toward Semantic Sensor Data Archives on the Web

Jean-Paul Calbimonte and Karl Aberer

EPFL, Switzerland
{name.surname}@epfl.ch

Abstract. Sensor datasets on the Web are becoming increasingly available, and there is a need for making them discoverable and accessible, so that they can be reused despite their heterogeneity. While RDF and Linked Data provide fundamental principles for sharing data on the Web, it is evidenced that they have limitations for efficiently transmitting and archiving sensor data. In this paper we identify some of the main challenges for engineering semantic sensor data archives, and we present an abstract architecture for such type of infrastructure. The proposed approach is based on a mix of RDF metadata and raw sensor archives with RDF mappings, so that data can be RDF-ized on demand. We use a real sensor deployment for air quality monitoring as a motivating use case and running example, and we show preliminary results on RDF transformation, compared with a representative data compression algorithm.

Keywords: Web Archive, Sensors, Internet of Things, Semantic archives

1 Introduction

The volume of sensor data on the web is growing at a fast pace, given the rise of the Web of Things and the increasing availability of data produced by wearable and mobile devices. Sensor datasets on the web may include weather and environmental measurements (e.g. MesoWest[1]), mountain and earthquake observations (e.g. USGS[2]), health monitoring readings, or geospatial sensing results (e.g. Swiss-Experiment[3]), among many others. These datasets can be reused for many different purposes. For example, air quality measurements can be used for training machine learning algorithms; or water pollution readings can be downloaded to compare with current levels on a different location. Other tasks include correlation computation and validation of measurements, or automatic calibration of sensor devices. Moreover, different sensor data sources can be integrated in a third-party application that produces added value on top of the original data. For instance, an application may use weather measurements, snow height observations and radiation levels to feed a mountain avalanche model simulation.

[1] Cf. http://mesowest.utah.edu/

[2] Cf. http://earthquake.usgs.gov/earthquakes/feed/v1.0/

[3] Cf. http://swiss-experiment.ch

Open data research has studied the problem of integration of web datasets in general. As a result, Semantic Web technologies have been shown to provide useful standards and mechanisms for sharing and reusing data on the Web. The Linked Data [1] principles provide a well-established blueprint for publishing data on the Web, and several examples of datasets are already available, following the RDF[4] and SPARQL[5] standards. These datasets include archives in different domains, and the so-called *LOD cloud* (`http://lod-cloud.net/`) includes sensor datasets, such as the LinkedSensorData[6] repository. However, these initiatives, commonly associated to the concept of the *Semantic Sensor Web* [14], only target preserving static datasets with little or no dynamics incorporated into the publishing and data access pipeline. Even if there are emerging efforts for adapting Linked Data for dynamic Web archiving [10], the usual guidelines for publishing RDF data on the Web do not always fit the characteristics of sensor data. More specifically, the generation of RDF from raw data or other data sources (e.g. using RDB2RDF[7] or other triplification engines) has the drawbacks of verbosity and data model complexity. An RDF representation of a sensor observation is typically composed of several triples, specifying the observation value, type of observed property, feature, sensing entity, time annotation, unit of measurement, etc. Ontology models such as the SSN (Semantic Sensor Networks) Ontology exemplify this complexity [5]. Even if this data representation approach provides a rich formalization and allows complex querying and reasoning, the volume of data can quickly explode and make it hard to transmit, share and archive sensor data.

In this paper we present some of the key challenges for managing sensor data archives, from a Semantic Web perspective. Then we propose a set of requirements and principles for building a semantics-enabled archive, which provides discovery and access to heterogeneous sensor observations stored in raw minimalistic formats and databases. These datasets can be made available through mappings that help transforming the sensor data into RDF on-the-fly. We argue that this approach can be convenient, compared to RDF materialization, or even compared with RDF compressed formats for archiving. In addition, we propose to combine this approach with database RDB2RDF [6] live transformation for archive queries and filters over the sensor data. To illustrate our study we present the example of a network of mobile sensors for air quality, which is intended to publish the collected data publicly, and provides a real-life use case for a sensor data archive. Finally, we discuss the issues that we found analyzing this topic, and the potential of our proposed approach. In order to support it, we provide preliminary experimentation results on real-life datasets.

The remainder of the paper is structured as follows. First we describe the challenges and requirements of semantic sensor data archives in Section 2. We present our use case about air quality monitoring in Section 3. Section 4 we

[4] Cf. `http://www.w3.org/TR/rdf11-primer/`
[5] Cf. `http://www.w3.org/TR/sparql11-overview/`
[6] Cf. `http://wiki.knoesis.org/index.php/LinkedSensorData`
[7] Cf. `https://www.w3.org/2001/sw/rdb2rdf/`

provide more concrete details about the principles and architecture of our archive approach. Section 5 contains a discussion about the potential of our approach, supported by preliminary experimentation. Section 6 concludes the paper.

2 Challenges in Sensor Data Archives

Sensor data is used in a very large range of applications, and the requirements in each of these are also numerous. Nevertheless, it is possible to identify basic requirements for those cases where sensor data needs to be preserved and archived for future access and re-use. This is common case in sensor data management, given that sensor data is usually only an initial input for further processing, which can be performed online (e.g. for alerts and real-time services) or offline (e.g. for batch processing and historical data analytics) [5]. Therefore, even if the sensor datasets have already been processed, they may be required for verification, replay, used as training data or for provenance checks and reproducibility of results. In the following, we describe some of the main aspects to consider in sensor data archives in this context, and the corresponding requirements and associated challenges.

Discoverability. This is a key aspect for sensor data archives, as they need first to be identified before being useful for any purpose. These include: discoverability, storage, reusability, accessibility, and interoperability. These datasets provide first-hand raw information about events in a particular domain of interest, but are of no use if the subject of sensing cannot be identified and searchable. In this respect, explicit semantics on the sensor metadata have shown to be useful, as they provide a common understanding of the objects of sensing, thanks to agreed models such as the SSN (Semantic Sensor Networks) Ontology [5], or similar formalizations. Following the SSN terminology, we can identify essential discovery criteria for sensor data, including:

- The observed property, or quality that is observed by the sensor, e.g. temperature, CO2 level, etc.
- The feature of interest, or object of which a quality is observed, e.g. the air at some location, a person, a room, etc.
- The sensor or type of sensor. Sensors can be physical or logical, they can be artificial or natural, or even abstract entities that capture observations. Types of sensors can be defined in terms of a taxonomy or also by other attributes inherents to the sensor (capabilities, model, etc.)
- The location, i.e. where the observations are made. Sensors can also be mobile, and therefore a location can be dynamic over time, but include a limited spatial coverage.
- The temporal extent, i.e. when the observations were made. This can be thought as time intervals or more complex time constraints that the archived observations should match.

Storage. Persistence of sensor data is not always required. In many applications sensor data is consumed live and only aggregations or specific events are stored

permanently. However, when either original or derived sensor data needs to be preserved, different archival options are available. In large sensor data archives there is a need for reducing volume as much as possible, using compressed formats if necessary. While in some cases storage can be provided through a database, this is not always the case as the querying and transactional requirements of these archives are often less critical than for live sensor data. In fact it is commonplace to find silos of sensor data in the form of compressed files. Choosing an adequate data format and compression mechanism is important for later stages as we will see afterwards. Other persistence considerations related to infrastructure such as replication or backup go beyond our analysis, but should not be disregarded.

Reusability. One of the main reasons for maintaining sensor data archives is the possibility of reusing the data for other purposes, potentially different for those for which they were originally collected. Archived sensor data can be very useful for different purposes. For instance for comparing sensor data from another location or from another sensor network of comparable characteristics. They can also be used for calibration purposes or for finding correlations. They can also be used for historical and batch analysis, for benchmarking or as training datasets for data mining algorithms. Another common sue case is for feeding numerical models that simulate physical phenomena. All these reusability use cases would not be possible if the data consumers are not able to determine the nature of the sensor data, i.e. a description of what is sensed, under which units, on what time range, locations, etc.

Accessibility. Data access should be possible through APIs and protocols that allow automation of data consumption from both people and software applications. The usage of de-referenceable URIs is a starting point for guaranteeing a simple but effective retrieval of sensor data. The SPARQL language can provide additional querying mechanisms that allow selecting relevant parts of the data declaratively. However, complex queries are not always required for slicing sensor data archives. In most cases simple time interval selections or similar filters are just enough, so it is often more useful to allow very simple access mechanisms on such datasets.

Interoperability & Standardization. The RDF and SPARQL standards are a building block for publishing data, and in our case, archives on the web. Specific ontologies and vocabularies, such as the Semantic Sensor Network (SSN) ontology can be used to represent both sensor metadata, and observations. Moreover, there exist domain specific ontologies that can be plugged to SSN, targeting, for instance, environmental monitoring, health sensing, manufacturing, etc. Other vocabularies have been standardized for describing a catalog[8], and can help for enabling the discoverability of sensor datasets.

[8] DCAT: `https://www.w3.org/TR/vocab-dcat/`

3 Air Quality Monitoring: A Use Case

We introduce a relevant use case for sensor data archives, which is centered on the area of air quality monitoring. Ambient air quality in the context of a city can be assessed in function of different pollutants present in the air, emitted and produced by different sources and having distinct characteristics, and of consequently different ways of being detected. Pollutants include, among others, Carbon monoxide (CO), Nitrogen dioxide (NO2), Nitrogen monoxide (NO), Sulfur dioxide (SO2), Ground level Ozone (O3), Particulate matter (PM) and Lead (Pb). These pollutants are produced by incomplete combustion processes, transportation, industrial and household combustion, and reactions of other pollutants. Considering the high impact of air quality, not only on the health and lifestyle of citizens, but also on the rest of the components of an urban settlement [16], its study has become a global concern. In fact, it is estimated that around 7 million premature deaths are attributable to air pollution [13] worldwide. For these reasons, air quality sensor data attracts the interest of interdisciplinary teams, with particularly different goals. Data consumers of these datasets may include public health researchers, governmental planning staff, environmental scientists, medical practitioners, traffic control agencies, etc.

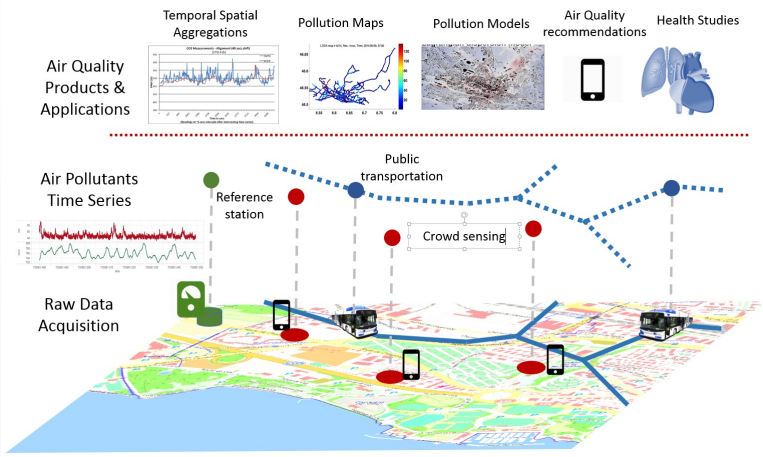

Fig. 1: The OpenSense2 approach for Air Quality monitoring in Smart Cities.

The OpenSense[9] project aims at integrating air quality measurements captured by heterogeneous mobile and crowdsensed data sources, in order to understand the health impacts of urban air pollution exposure, and providing high-resolution air quality maps. Urban air quality requires a complex deployment of sensors, as the one we propose in OpenSense, composed of devices mounted on public transportation, and personal mobile sensors (Figure 1). OpenSense allows not only monitoring the concentration of pollutants, but also letting this information be visible and accessible publicly, so that citizens can learn and adapt their behavior to the measured conditions. This data management life-cycle, is

[9] OpenSense project: `http://opensense.epfl.ch`

orchestrated by the GSN (Global Sensor Networks [3]) middleware, from data acquisition to publishing and data access.

The data collection from the sensors produces time series of the air pollutant measurements, which are only the first step in the OpenSense pipeline. The resulting dataset consists of raw observations that cannot be directly used by citizens and external applications. To understand the semantics of this data, additional processing is required, exploiting first the spatial characteristics of the data points. Given the highly localized nature of air pollutant concentrations (due to physical behavior characterized in models such as street canyons), we need to project the observations to street segments and perform spatial aggregation and interpolation. Once the spatio-temporal distributions have been computed and made available, pollution maps can be generated, complemented or validated with this data, leading to more advanced air quality models: e.g. log-linear regression models, lagrangian dispersion models, etc. Finally, using these models, end-user applications can be built leveraging on the processed datasets.

In sum, we can see that OpenSense introduces the need to deal with a large number of sensing devices, which are potentially very different in terms of capabilities (e.g. accuracy, resolution, precision, reliability, etc.). Furthermore, there is a need to access and process their data not online live, but also offline, given the numerous potential uses of the resulting datasets. Also, these datasets can be potentially integrated in order to gain more insights or combine with higher level models. Given the large number of sensors and large volumes produced by each of them, it is certainly necessary to transmit, store and share these datasets s efficiently as possible. As we can see, most of the requirements previously stated apply to an archive based on the data produced by the OpenSense project.

4 A Semantically-enabled Sensor Data Archive

In this section we present the main design principles for a sensor data archive that incorporates semantic data management at its core, in order to cope with some of the requirements described previously in Section 2.

4.1 Design Principles

Based on the challenges and requirements identified in Section 2, and our experience in previous sensor data monitoring systems and projects (e.g. SwissExperiment, OSPER, OpenSense, OpenIoT (`http://openiot.eu`), D1namo[10]), we advocate for an archiving approach that exploits semantic sensor metadata and raw sensor data, under the following considerations.

Sensor data regularity: Raw sensor data is typically collected as time series of data items, presenting in most cases a very regular structure. Consider the example of mobile NO2 sensor readings in Listing 1. These are series of tuples that contain a timestamp, the sensor identifier, the observed value and the geographical coordinates:

[10] Cf. `http://www.nano-tera.ch/projects/456.php`

```
29-02-2016T16:41:24,47,369,46.52104,6.63579
29-02-2016T16:41:34,47,358,46.52344,6.63595
29-02-2016T16:41:44,47,354,46.52632,6.63634
29-02-2016T16:41:54,47,355,46.52684,6.63729
...
```

Listing 1: Example of mobile NO2 sensor readings.

The regularity of these datasets is a a common characteristic that can be exploited for archiving purposes. Compact representations, such as CSV for human-readable purposes, can be used fairly easily, allowing users to share and reuse these datasets and load them into analytic infrastructures or visualization tools.

Furthermore, this regularity is extremely useful for data compression purposes, which can make data storage and interchange faster and more efficient. Nevertheless, this compact representation lacks minimal information to be useful by third parties.

As we will see later we propose to associated semantic metadata and RDF mappings to interpret and understand these time series.

Sensor data order: In these archives the order of sensor data is crucial. Given that these come from sensor streams, time is typically the key attribute for establishing an order among the data items. This is important for indexing the data items, allowing for fast insertions in append mode, which are generally optimized in database and file storage systems. This also enables efficient time-based selection, filtering and windowing over the data, which are common operators in this context.

Sensor metadata: The metadata is essential in order to allow discoverability and self-description of sensor datasets. The metadata includes not only information about the sensor and its characteristics, but also meta-information about the generated datasets and how they can be accessed and exploited. The usage of existing vocabularies is critical at this level, and the usage of standards is recommended. First, for the dataset metadata we propose using the DCAT vocabulary, which allows defining catalogs and datasets. For instance, a sensor catalog for OpenSense can be specified as follows:

```
:sensorCatalog a dcat:Catalog ;
       dct:title "OpenSense data catalog" ;
       dct:language iso639-1:en ;
       dct:publisher :LSIR-EPFL ;
       foaf:homepage <http://opensense.epfl.ch/data/> ;
       dcat:dataset :geo-osanm, :geo-osfpm , :geo-oso3m .
```

Listing 2: Example of sensor catalog metadata in OpenSense.

A catalog may include several datasets, e.g. one dataset can represent one time series produced by a sensor. A dataset can also group several sensor readings, especially if these are related to the same observed property or the same geographical space. These aggregations can be configured as virtual sensors. As an example, consider a NO2 dataset produced by a virtual sensor that aggregate

data from several sensor boxes. The dataset metadata can include the temporal and spatial extent. Furthermore the dataset metadata can be combined with the SSN ontology, in order to include specific information about the sensor, such as measurement capabilities, calibration, etc. (Listing 3).

```
:geo-osanm a dcat:Dataset;
      dct:title "OpenSense NO2 measurements";
      dcat:theme :NO2;
      dct:issued "2015-12-05"^^xsd:date;
      dct:temporal g-interval:1977-11-01T12:22:45/P1Y;
      dct:spatial <http://www.geonames.org/6695072>;
      dct:publisher :LSIR-EPFL;
      dct:accrualPeriodicity sdmx:freq-W;
      ssn:isProducedBy :NO2VsensorBox;
      dcat:distribution :geo-osanm-csv .

:NO2VsensorBox a ssn:Sensor;
      rdfs:label "NO2 Virtual Sensor Lausanne";
      ssn:observes :NO2;
      ssn:hasMeasurementCapability [
         a ssn:Accuracy;
         ssn:forProperty :NO2;
         ssn:inCondition ... ;
         ssn:hasValue ... ] .
```

Listing 3: Example of metadata for an OpenSense NO2 dataset.

Finally, for each dataset we may have different distributions. In our case, we can expose the sensor observations as CSVs, which can be represented as follows:

```
:geo-osanm-csv a dcat:Distribution ;
      dcat:downloadURL <http://opensense.epfl.ch/data/api/sensors/geo_osanm>;
      dct:title "CSV distribution of NO2 measurements";
      dcat:mediaType "text/csv";
      dcat:byteSize "5534530"^^xsd:decimal .
```

Listing 4: Example of a Dataset Distribution metadata.

Semantically-enabled sensor data: While the sensor data archive chiefly serves raw data accompanied with semantic sensor metadata, it may be needed to provide a semantically-enabled version of the data, so that it can be further integrated, processed or reasoned upon. In order to do so we propose using R2RML mappings (and related extensions) to transform the compact raw sensor data into semantics-rich RDF datasets. While we could think of storing materialized RDF datasets, this may result not too convenient for the following reasons. First, data may be updated periodically, and therefore we would need to maintain a synchronization service that permanently runs this conversion, taking up additional resources. Second, the RDF versions of the dataset can grow up to multiple orders of magnitude with respect to the original dataset. This verbosity can quickly overload the archive resources, and it can be cumbersome for data interchange. Third, given that sensor data present recurring patterns, the RDF representation can be unnecessarily repetitive (e.g. the same unit of measurement, same feature of interest, same observed property, making it harder for data consumers to focus on the essential information contained in the archive. Finally, a materialized RDF dataset would be too rigid if the data consumers need to use a different ontology or vocabulary for the same dataset. It may be

required to use different ontologies for integration purposes, or for focusing on different aspects of the dataset (e.g. using a statistics ontology for some tasks, or a provenance ontology for others).

Bulk data access: The bulk data download should enable not only the access to the raw data, but also to alternative formats including JSON (`http://json.org/`), NetCDF (`http://www.unidata.ucar.edu/software/netcdf/`), or other available options, which may vary depending on the domain of discourse. Nevertheless, when the data is required to be transformed to RDF, we propose using the SSN ontology, and more specifically the observation module. The usage of the SSN ontology is appropriate as it is domain-agnostic [5] and has shown wide adoption and application in different scenarios. An example of an NO2 observation, including a value, result time and other details can be encoded as follows:

```
:no2obs1 a :NO2Observation ;
      ssn:observedProperty :NO2 ;
      ssn:featureOfInterest aq:AirMedium ;
      ssn:observedBy :NO2SensorBox ;
      ssn:observationResult :no2obs1result ;
      ssn:observationResultTime :instant_20160331232000 .

:no2obs1result a :NO2ObservationValue ;
      qu:numericalValue "345.00"^^xsd:float ;
      qu:unit :ppm .

:instant_20160331232000 a time:Instant ;
      time:inXSDDateTime "2016-03-31T23:20:00"^^xsd:datetime .
```

Listing 5: Example of an NO2 observation in RDF using the SSN ontology.

Time-based selection & filtering: Although one of the main tasks in the proposed sensor data archive is related to the direct download of a dataset, it is also crucial to provide minimal selection features, especially given that users are typically only interested in a part of the dataset, rather than the full version. The most commonly requested type of selection is based on time filters, which establish temporal bounds, defined as an interval of two-timestamps. Other special cases include gathering the latest values, a particular point in time, an interval defined with an offset, or even seasonal and periodic queries. Given that the archive is ordered, and potentially indexed with respect to time, these operations should be optimized and supported out of the box.

Additional operations such as basic temporal analysis and simple anomaly detection are also commonly required in a sensor archive [4, 17]. While such additional considerations are relevant, for the sake of simplicity we do not expand on these and other related topics.

4.2 Architecture

We propose an abstract architecture (Figure 2) for a Semantic Sensor Data Archive, which includes features that answer to the requirements above, and follows the principles described previously.

First, in terms of storage we envision three main storage components. One targets long term stored datasets, in formats such as CSV (although other formats can also be included). The second is a database (the underlying technology

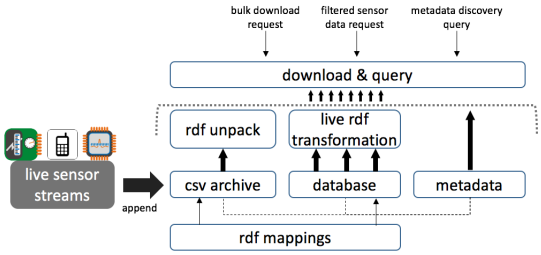

Fig. 2: Semantic Sensor Data Archiving architecture.

of this database could be of different kinds) which is used for those datasets where more flexible filtering and selection policies are allowed. This can be of interest for very large datasets where the complete body of data is too large an unnecessary for consumers. It is perfectly possible that the stored datasets can be loaded into the database store (e.g. load a CSV into a database). Therefore, a transition between these two is a possibility. The third an final storage component in the metadata store, which contains the dataset annotations in RDF, as described in the example above.

In addition to these stores we propose a mappings repository that stores the R2RML-extended mappings that specify how to produce live RDF from the long term storage datasets and the database. In the following example (Listing 6) we present a sample pair of R2RML mappings that aim at producing instances of NO2 Observations and their corresponding Observation values. The first mapping defines how the observation value is produced. Its URI is defined using a template for the `subjectMap`, which replaces the `sensor` and `time` column values at runtime to produce the final URI value. Then the observation numerical value – i.e. the actual observed NO2 value– is mapped through a `predicateObjectMap` with a fixed predicate `qu:numericalValue`. The unit of measurement is fixed through another R2RML mapping not depicted here, for space constraints.

```
:ObsValueMap
  rr:subjectMap [
    rr:template "http://opensense.epfl.ch/data/ObsResult_NO2_{sensor}_{time}"];
  rr:predicateObjectMap [
    rr:predicate qu:numericalValue; rr:objectMap [ rr:column "no2"; rr:datatype xsd:float; ]];
  rr:predicateObjectMap [
    rr:predicate obs:uom; rr:objectMap [ rr:parentTriplesMap :UnitMap; ]].
:ObservationMap
  rr:subjectMap [
    rr:template "http://opensense.epfl.ch/data/Obs_NO2_{sensor}_{time}"];
  rr:predicateObjectMap [
    rr:predicate ssn:observedProperty;
    rr:objectMap [ rr:constant opensense:NO2]];
  rr:predicateObjectMap [
    rr:predicate ssn:observedBy;
    rr:objectMap [ rr:template "http://opensense.epfl.ch/data/Sensor_{sensor}"]];
  rr:predicateObjectMap [
    rr:predicate obs:result; rr:objectMap [ rr:parentTriplesMap :ObsValueMap]].
```

Listing 6: Example of an R2RML mapping for an OpenSense NO2 dataset.

For the observation itself, the `ObservationMap` mapping follows a similar structure. Again, the URI is constructed with a template that uses the sensor and time columns. It specifies the observed property, using a fixed URI, and the observing sensor using a URI template. The Observation value is linked through a `parentTriplesMap` reference. More information on R2RML is available in [6].

While these mappings can be conveniently used for on-demand transformation to RDF of sensor data in bulk and filtering data access, RDF is oftentimes not the most convenient representation for data sharing and transmission. With these considerations in mind, we opt for a commonly use format for sensor observations as CSV, but augmented with rich semantic descriptions that follow the specifications of the CSV on the Web Working Group[11]. As an example of embedded metadata that can be provided for the OpenSense data sets, consider the JSON snippet below (Listing 7). It represents a description of a CSV output of sensor data from the GSN middleware, using the metadata model defined by the CSV on the Web group [15].

```
{
  "@context": ["http://www.w3.org/ns/csvw",
               {"@language": "en",
                "base": "http://opensense.epfl.ch/data/",
                "time": "http://www.w3.org/2006/time#",
                "ssn": "http://purl.oclc.org/NET/ssnx/ssn#",
                "qu": "http://purl.oclc.org/NET/ssnx/qu/qu#",
                "opensense":"http://opensense.epfl.ch/onto/opensense#"}               ],
  "tableSchema": {
    "columns": [{
      "name": "time",
      "titles": "Timestamp",
      "aboutUrl": "Instant_{time}",
      "propertyUrl": "time:inXSDDateTime",
      "datatype": {"base": "dateTime","format": "yyyy-MM-ddTHH:mm:ss" }
    }, {
      "name": "sensor",
      "titles": "Bus sensor",
      "aboutUrl": "Obs_NO2_{sensor}_{time}",
      "propertyUrl": "ssn:observedBy",
      "valueUrl": "Sensor_{sensor}"
    }, {
      "name": "obsProperty",
      "virtual": true,
      "aboutUrl": "Obs_NO2_{sensor}_{time}",
      "propertyUrl": "ssn:observedProperty",
      "valueUrl": "opensense:NO2"
    }, {
      "name": "obsResult",
      "virtual": true,
      "aboutUrl": "Obs_NO2_{sensor}_{time}",
      "propertyUrl": "ssn:observationResult",
      "valueUrl": "ObsResult_NO2_{sensor}_{time}"
    }, {
      "name": "obsTime",
      "virtual": true,
      "aboutUrl": "Obs_NO2_{sensor}_{time}",
      "propertyUrl": "ssn:observationResultTime",
      "valueUrl": "Instant_{time}"
    }, {
      "name": "no2",
      "titles": "NO2 concentration",
      "aboutUrl": "ObsResult_NO2_{sensor}_{time}",
```

[11] CSV on the Web `http://www.w3.org/TR/csv2rdf/`

```
    "propertyUrl": "qu:numericalValue"
  }, {
    "name": "unit",
    "virtual": true,
    "aboutUrl": "ObsResult_NO2_{sensor}_{time}",
    "propertyUrl": "qu:unit",
    "valueUrl": "opensense:ppm"
  }]  }
```

Listing 7: Example of a JSON metadata for a CSV of an OpenSense NO2 dataset.

The (simplified) example provides an explicit description of the columns of a CSV table consiting of NO2 observations in OpenSense. Following the CSV2RDF conversion standards of the CSV on the Web recommendations, RDF can be generated as follows. First, the `time` column will generate a triple, whose subject is the observation IRI, that will be linked to the date-time through the `ssn:observationResultTime`. Similarly, the `sensor` column defines metadata that links to the URI of the sensor that produced the observation. The definition in the `no2` column will generate a relationship with the observation value. In addition, the CSV on the Web specification allows defining virtual columns, which are useful to create more than one subject per CSV row (e.g. for the units, and for the observation type).

This approach can be interesting for data consumers that are not necessarily familiar with Semantic Web standards, as it comes in the form of a seemingly ordinary CSV. Nevertheless, thanks to the CSV-to-RDF standard transformations encoded in the metadata, it is easier to interpret the dataset as a set of RDF triples. An important issue is related to the capability of current RDF triple stores to handle streams of very dynamic RDF data. While it is certainly technically possible to store and archive data in such databases [3], it remains a challenge to perform continuous processing and stream data querying over RDF Streams. The ongoing work at the W3C RDF Stream Community Group[12], and the current prototypes that follow this new processing paradigm can use the stream of data produced by the OpenSense deployment as an input for more advanced continuous query processing.

5 Discussion

The presented architecture and approach for a Semantic Sensor Data archive considers some of the basic requirements for such an infrastructure. One of this aspects is related to the efficiency of transmission and interchange. A possible alternative to our proposal could be to use RDF as a representation format all the way down to the storage level. The advantage of this would be that no mappings would be needed in order to transform from raw data sources such as databases and CSV files. Furthermore, users could directly query these data sources with SPARQL. However, as we discussed previously, for the typical archival use cases a complex query language such as SPARQL is not essential for data retrieval, and even less for bulk data acquisition. In addition, sensor observation data is

[12] W3C RSP Community Group: `http://www.w3.org/community/rsp`

commonly better understood in simple tabular and time-series formats, rather than in graph-based structures such as RDF.

Another important aspect is size and storage. As stated above, sensor observation data can be huge ad efficiency for transmission and interchange is vital. RDF, in its many serializations remains too verbose, even for very simple datasets such as observation values. Compression mechanisms exist to overcome this issue, which have shown to greatly reduce the amount of bytes to store graphs and triples [7]. These includes among others, binary representations such as HDT [9], or stream-aware approaches such as RDSZ [11] and ERI [8]. Compression algorithms such as ERI take advantage of the repeating nature of a dataset. This is a common property on most sensor observation data sources, as we saw previously, and therefore these algorithms could be used in our context. We have developed a proof of concept implementation of our approach for representing sensor observations as CSV, including mappings that allow transforming the dataset into RDF triples.

We have compared our approach with the optimized ERI interchange format for RDF. As a preliminary evaluation, we used the Linked Sensor Data [14] datasets, most specifically the Nevada dataset that contains observations about environmental properties during blizzard and hurricane events. As we can see in Figure 3(a), for 38 million triples, the transformation of the compressed CSV format+mappings, into RDF, is reduced from around 8GB to 214MB. Compared to the ERI representation (147MB) our approach offers a very competitive size, which is not even binary-compressed and is directly human-readable. Furthermore, if we apply a simple zip compression on top of our approach we get an even more compact pack of only 19MB. In terms of decompression time, our approach throws similar execution time as ERI, as we can see in Figure 3(b).

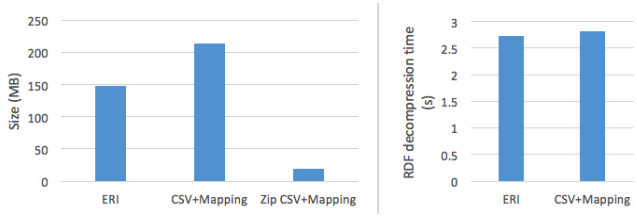

Fig. 3: OpenSense Archiving. Comparison in space (a), and execution time (b) with ERI [8].

While these results apply for bulk transfer and interchange, we are also interested in how this would result for query filtering. We have performed preliminary experiments, running the on-demand RDF transformation on top of a Postgres relational database containing the Nevada Linked Sensor Data. The filtering data results in Figure 4 show that the RDF transformation time has a linear growth as the number of observation points increases.

Data format transformation can potentially become an issue, if the heterogeneity of sources is not taken into account. CSV is not the only possible input

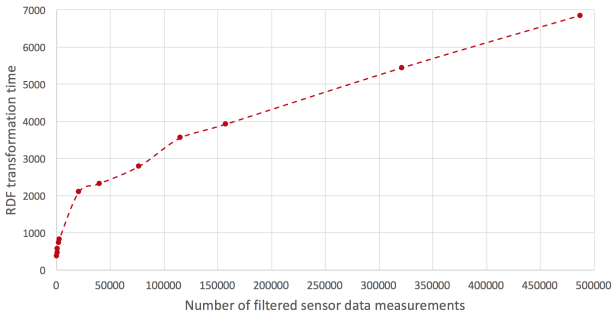

Fig. 4: RDF transformation time for sensor data filtering.

format for RDF streams, and optimizations to efficiently transform data online or in batch mode are essential. Related solutions such as XSPARQL [2] or live RDF Stream generation [12] can be complementary in this context.

6 Conclusions & Future Work

We have presented an architecture for sensor data archives, which incorporates Semantic Web standards at its core, in order to enhance discoverability, accessibility, interoperability and data sharing. We advocate for the use of hybrid sensor data storage, including RDF metadata for sensor description, using standards based on the SSN ontology. Furthermore, we consider using simple formats for sensor observations data exchange, such as CSV, augmented with semantic annotations in JSON,following the W3C Recommendations for CSV on the Web. Finally, we have scoped our work and presented it as part of a real sensor data archive for Air Quality Monitoring, which is expected to implement the proposed architecture.

In the future we plan to complete and showcase the semantic-aware features of our sensor data archive in OpenSense, comparing it to alternative approaches for data archiving and preservation. We plan to investigate on further optimizations regarding sensor data compression, including value encoding and approximation, or pre-compilation of mapping-based transofrmations. Furthermore, we plan to design and publish open-source sensor data crawlers that will be able to exploit the sensor metadata exposed by this type of archives.

Acknowledgments Partially supported by the Nano-Tera.ch OpenSense2 project, evaluated by the Swiss National Science Foundation.

References

1. Berners-Lee, T., Hendler, J.: Publishing on the semantic web. Nature 410(6832), 1023–1024 (2001)

2. Bischof, S., Decker, S., Krennwallner, T., Lopes, N., Polleres, A.: Mapping between rdf and xml with xsparql. Journal on Data Semantics 1(3), 147–185 (2012)
3. Calbimonte, J.P., Sarni, S., Eberle, J., Aberer, K.: Xgsn: An open-source semantic sensing middleware for the web of things. In: Proc. of the 7th International Workshop on Semantic Sensor Networks (2014)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3), 15 (2009)
5. Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C semantic sensor network incubator group. Journal of Web Semantics 17, 25–32 (2012)
6. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. https://www.w3.org/TR/r2rml/ (2012)
7. Fernández, J.D., Gutierrez, C., Martínez-Prieto, M.A.: Rdf compression: basic approaches. In: Proceedings of the 19th international conference on World wide web. pp. 1091–1092. ACM (2010)
8. Fernández, J.D., Llaves, A., Corcho, O.: Efficient rdf interchange (eri) format for rdf data streams. In: The Semantic Web–ISWC 2014, pp. 244–259. Springer (2014)
9. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary rdf representation for publication and exchange (hdt). Web Semantics: Science, Services and Agents on the World Wide Web 19, 22–41 (2013)
10. Fernández, J.D., Polleres, A., Umbrich, J.: Towards efficient archiving of dynamic linked open data. Proc. of DIACHRON pp. 34–49 (2015)
11. Fernández, N., Arias, J., Sánchez, L., Fuentes-Lorenzo, D., Corcho, Ó.: Rdsz: an approach for lossless rdf stream compression. In: The Semantic Web: Trends and Challenges, pp. 52–67. Springer (2014)
12. Mauri, A., Calbimonte, J.P., Balduini, M., Della Valle, E., Aberer, K., et al.: Where are the rdf streams?: Deploying rdf streams on the web of data with triplewave. In: 14th International Semantic Web Conference ISWC (2015)
13. Organization, W.H.: News release. http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/. (March 2014)
14. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. Internet Computing, IEEE 12(4), 78–83 (2008)
15. Tennison, J., Kellogg, G., Herman, I.: Model for tabular data and metadata on the web. http://www.w3.org/TR/tabular-data-model/ (2015)
16. Tsai, D.H., Guessous, I., Riediker, M., Paccaud, F., Gaspoz, J.M., Theler, J.M., Waeber, G., Vollenweider, P., Bochud, M.: Short-term effects of particulate matters on pulse pressure in two general population studies. Journal of hypertension 33(6), 1144–1152 (2015)
17. Vuran, M.C., Akan, Ö.B., Akyildiz, I.F.: Spatio-temporal correlation: theory and applications for wireless sensor networks. Computer Networks 45(3), 245–259 (2004)