The Linked Sensor Middleware — Connecting the real world and the Semantic Web

Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Josiane Xavier Parreira, and Manfred Hauswirth

Digital Enterprise Research Institute – National University Ireland
Galway, Ireland
{danh.lephuoc,hoan.quoc,josiane.parreira,manfred.hauswirth}@deri.org

Abstract. Sensing devices are becoming the source of a large portion of the Web data. To facilitate the integration of sensed data with data from other sources, both sensor stream sources and data are being enriched with semantic descriptions, creating Linked Stream Data. Despite its enormous potential, little has been done to explore Linked Stream Data. One of the main characteristics of such data is its "live" nature, which prohibits existing Linked Data technologies to be applied directly. Moreover, there is currently a lack of tools to facilitate publishing Linked Stream Data and making it available to other applications.
To address these issues we have developed the *Linked Stream Middleware* (LSM), a platform that brings together the live real world sensed data and the Semantic Web. A LSM deployment is available at http://lsm.deri.ie/. It provides many functionalities such as: i) wrappers for real time data collection and publishing; ii) a web interface for data annotation and visualisation; and iii) a SPARQL endpoint for querying unified Linked Stream Data and Linked Data. In this paper we describe the system architecture behind LSM, provide details how Linked Stream Data is generated, and demonstrate the benefits of the platform by showcasing its interface.

Keywords: Sensor mashup, live linked stream data, semantic sensor data, Linked Data query processing

1 Introduction

Sensing devices have become ubiquitous. In 2005, Gartner predicts that *"By 2015, wirelessly networked sensors in everything we own will form a new Web. But it will only be of value if the 'terabyte torrent' of data it generates can be collected, analysed and interpreted."*[1] Making sensor generated streams of data usable as a new and key source of knowledge requires its integration into the existing information space of the Web.

[1] Mark Raskino, Jackie Fenn, and Alexander Linden. Extracting Value From the Massively Connected World of 2015. Gartner Research, 1 April 2005. http://www.gartner.com/resources/125900/125949/extracting_valu.pdf

Recently, there has been efforts to lift stream data to a semantic level, e.g., by the W3C Semantic Sensor Network Incubator Group[2] and [3, 7, 8]. The goal is to make stream data available according to the Linked Data principles [2] – a concept known as *Linked Stream Data* [6], which allows an easy and seamless integration, not only among heterogenous sensor data, but also between sensor and Linked Data collections, enabling a new range of "real-time" applications.

Despite its enormous potential, little has been done to explore Linked Stream Data and connect it with other data sources, such as the ones found in the Linked Open Data cloud. This is mainly due to the fact that Linked Stream Data has a temporal, "live" aspect that is not present in Linked Data, which prohibits existing Linked Data technologies to be applied directly. Moreover, there is currently a lack of tools to facilitate publishing Linked Stream Data and making it available to different applications, either directly from the data source or via query interfaces.

To address these issues we have developed the *Linked Stream Middleware* (LSM), a platform that brings together the live real world sensed data and the Semantic Web in an unified model. The LSM provides an extensive range of functionalities: different wrappers are used to access stream sources and transform the raw data into Linked Stream Data; data annotation and visualisation is possible via an easy to use web interface; and live querying over unified Linked Stream Data and data coming from the Linked Open Data cloud is enabled by two types of query processors – a standard Linked Data query processor and CQELS [5], a query processor for handling both Linked Stream Data and Linked Data.

A LSM deployment is available at `http://lsm.deri.ie/`. In this paper we describe the system architecture behind LSM, provide details how Linked Stream Data is generated, and demonstrate the benefits of the platform by showcasing its interface. The paper also contains a small section about some of the lessons learned during the implementation of LSM and some challenges that remain open, and an Appendix that addresses the Semantic Web Challenge Requirements.

2 Exploring and streaming live data with LSM

A deployment of LSM is available at `http://lsm.deri.ie/`, where an user friendly interface allows sensor data to be published, visualised, annotated and queried. Figure 1 provides a screenshot of LSM, where the highlighted features are explained below.

The interface uses a map overlay to display the sensor information. Several types of sensor data are available, such as flight status (1), weather (2), trains/buses arriving times, nearby metro stations (3), street cameras (4), etc. We currently have access to live stream data from over 100,000 sensors around the world. The history of the data produced by a particular source can also be seen and downloaded in RDF format. Figure 1 shows an example of data history for a temperature sensor (items (2a) and (2b)).

Besides the map overlay, the data can also be accessed via an interactive faceted search feature. The faceted search allows users to filter the information displayed based on sensor type, given by a sensor taxonomy, on sensor location, by choosing an area in the map, or on sensor specification (sensor type, physical context, accuracy, etc).

[2] `http://www.w3.org/2005/Incubator/ssn/`

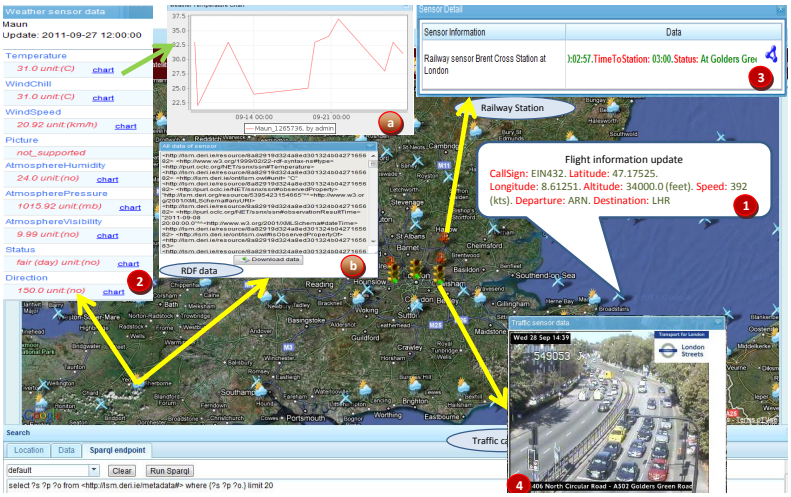

Fig. 1. Some of the features of LSM.

If a user has a sensor source that he would like to publish as Linked Stream Data, LSM provides an easy way to do so. Figure 2 demonstrate the process of annotating sensor data from a source in XML format. After the user chooses the type of the sensor to be added (step 1), LSM parses the input source to extract the properties (step 2), which can then be selected and annotated (step 3). Step 4 allows users to add extra sensor descriptions. Moreover, users can also import external ontologies into the middleware, to link the available data back to concepts of their domain.

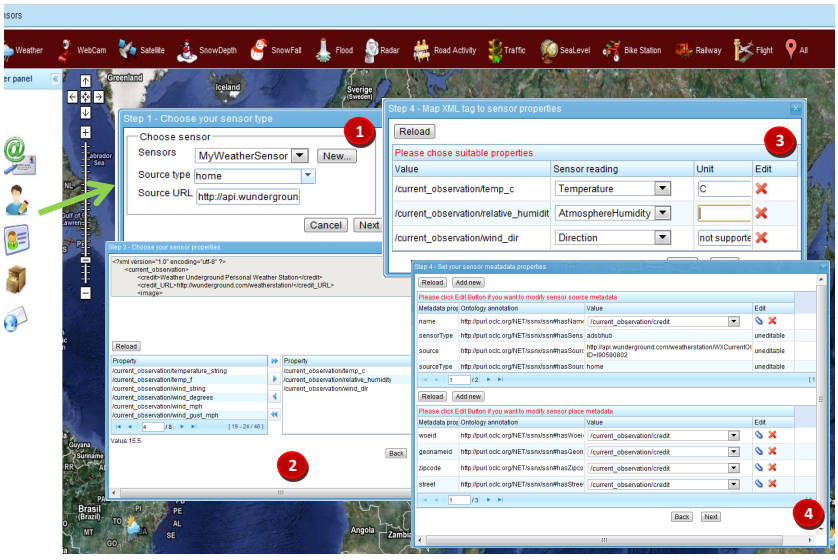

Fig. 2. Sensor data annotation.

For more advanced users, LSM also provides a SPARQL endpoint. In addition, LSM provides tools to create notifications or sensor feeds from live data sources. An example is given in Figure 3, where a sensor feed is expressed in CQELS [5], a query

processor that allows continuous queries over unified Linked Stream Data and Linked Data. The feed created can be easily fed into any application. To demonstrate the simplicity of building applications with sensor feeds, we created an overlay real-time sensor data application based on the location given by the user's mobile device. The application runs on the Android platform and contains only a few lines of code. The source code and application can be downloaded at `http://code.google.com/p/deri-lsm/`.

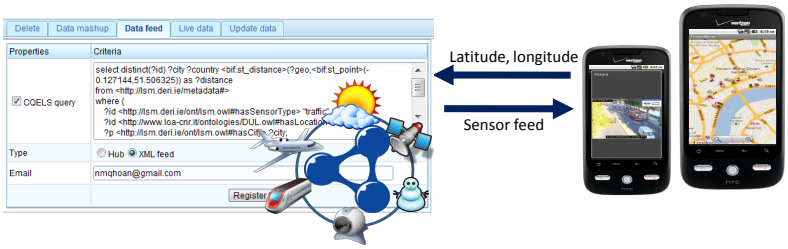

Fig. 3. Example of a sensor data feed for a mobile device.

3 Making sensor data *linkable*

Sensors generate raw data, which are heterogenous and can not be easily integrated with data from other sources. To make sensor data composable we provide global identities to their data items following the Linked Data publishing principles [4]. To increase the composability of these *linkable data items*, we represent the sensor data in a layered graph layout as shown in Figure 4.

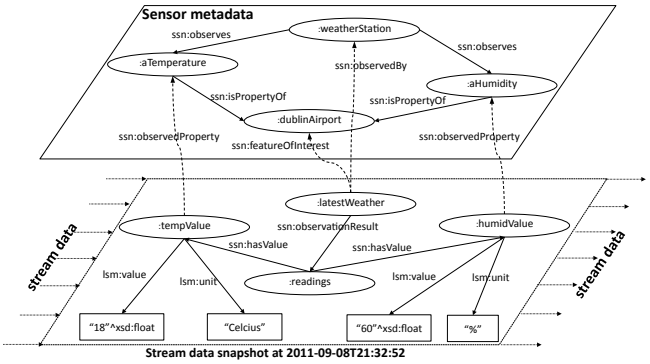

Fig. 4. Layered graph-based layout for Linked Stream Data.

Based on the Semantic Sensor Network (SSN) ontology[3] we define a static and a dynamic layer. The static layer describes the sensor metadata which is time independent, like sensor specification, information about objects observed by sensors, etc. For instance, Figure 4 describes a "***weather station*** that *observes* the **temperature** and **humidity** at ***Dublin Airport***". The dynamic layer contains a graph-based stream data from the time-varying sensor readings. These readings have links to their meanings, e.g."*tempValue(18 Celcius)* is the **temperature** of ***Dublin Airport*** at *21:32:52, 09/08/2011*". The layout serves as guideline for the annotation process that generates the data transformation rules for the wrappers.

[3] `http://purl.oclc.org/NET/ssnx/ssn`

4 Architecture and Implementation details

The LSM architecture is illustrated in Figure 5. It is divided in layers that together cover the entire process, from data acquisition, to Linked Data publishing and access, until applications. Next, we describe in detail the implementation of each of these layers.

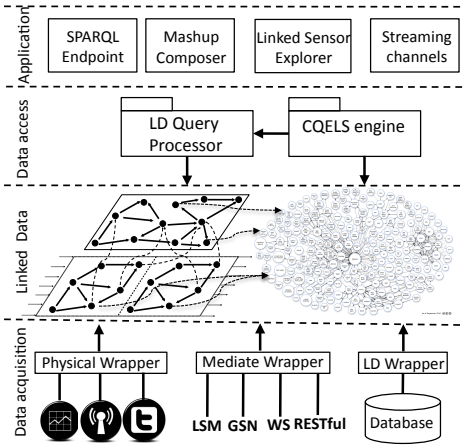

Fig. 5. Layered architecture of the LSM platform.

4.1 Data Acquisition Layer

The data acquisition layer provides wrappers to collect sensor readings and transform them according to the Linked Stream Data layout previously described. LSM currently provides three types of wrappers:

Physical wrappers: are used for collecting sensor data coming from physical devices via physical connections, for instance, serial and ad-hoc network connections.

Mediate wrappers: mediate the connections to other sensor middlewares by transforming data from a variety of data formats and data feeding protocols into RDF. LSM currently provides mediate wrappers to some of the most used middlewares like GSN [1], Pachube[4], the sensor gateway/web services (NOAA[5]), and the London Transport syndication[6].

Linked Data wrappers: provide access to the sensor data sources being collected and stored in relational databases by exposing the relational data structure in RDF via mapping rules like the ones provided in the D2R[7] mapping language.

4.2 Linked Data Layer

The Linked Data layer provides the interface to access not only Linked Sensor Data which was collected and annotated by the wrappers provided, but also outgoing links to

[4] http://pachube.com
[5] http://www.noaa.gov/
[6] http://www.tfl.gov.uk/businessandpartners/syndication/default.aspx
[7] http://www4.wiwiss.fu-berlin.de/bizer/d2rmap/d2rmap.htm

data in the Linked Data Cloud. The outgoing links are provided by users via the annotation module. LSM also automatically extracts relevant links from DBpedia, Geonames and LinkedGeoData via spatial relationships, for example, point of interests nearby a sensor location. In the current implementation, we use Virtuoso as the triple storage, the relational database and the spatial indexer for this layer.

4.3 Data Access Layer

The Data Access layer supports the declarative queries on top of the Linked Data layer. Queries over Linked Stream Data can follow either a pull-based or a push-based model. LSM provides two query processors, one for each of the query models, as described below. These query processors also enable data access from remote SPARQL endpoints via a federation extension of the SPARQL 1.1 language[8].

Linked Data query processor: supports traditional pull-based queries under SPARQL language over sensor metadata and sensor readings (live and historical).

CQELS engine: supports push-based continuous queries over Linked Stream Data under CQELS language [5]. This query engine allows user to actively filter and integrate real-time data to create new streams of data from existing sources.

4.4 Application layer

The query processing capability provided by the Data Access layer allows the easy and rapid application development for end-users or machine-users. In addition, the Application layer offers the following extra functionalities:

SPARQL Endpoint: makes the data available in LSM accessible as a citizen of the Linked Data Cloud.

Mashup composer: enables the composition of existing data sources to derive new sensor data sources. The derived data source is created via either a visual wizard or a continuous query under the CQELS language.

Linked Sensor Explorer: enables the exploration of existing data sources. A faceted browsing functionality helps the user to filter sensor data based on relevant properties, e.g. location and meanings of readings. In the background the exploration and navigation actions are translated into complex queries and executed on top of the Linked Data query processor, but all are done transparently to the user.

Streaming channels: actively stream integrated data sources via streaming protocols like Google PubSubHubbub and XMPP[9].

5 Lessons learned and Challenges

In the process of implementing the Linked Stream Middleware we faced with a number of design decisions and challenges. We discuss some of them here, since they might be relevant to other systems as well.

Scalability is a crucial feature for the targeting deploying scenarios of LSM. There are literally over millions of sensor data sources published via HTTP. We currently

[8] `http://www.w3.org/TR/sparql11-federated-query/`
[9] `http://code.google.com/p/pubsubhubbub/`, `http://xmpp.org/`

handle over 100,000 data sources, and for that the data fetching process needs to run in a reliable, scalable and bottleneck free infrastructure. We choose Hadoop for scheduling the data fetching and transformation performed by the wrappers. The main advantage is that it allows us to increase the data consumption and transformation rate by simply adding more hardware. However, how to adapt the scheduling to the variations on the stream rates and server throughputs is still an open challenge.

Besides processing real time data, it is also necessary to store the data generated, either for queries defined over a time period or for archiving purposes. We have observed that most of the triple storages can not efficiently handle high update rates. In addition, materialising sensor readings into triples is also inefficient, especially numeric readings. Therefore, we use relational tables to store historical data. However, this requires mapping the data from relations to triples. Virtuoso provides a mapping language as well as a query rewriter that enables querying relational data from SPARQL in a transparent way. However, we ran into performance issue with complicated queries, because the rewritten queries were too complex. Hence, better mapping languages and optimised query rewriters are still needed to deal with the scale and schema of the data.

References

1. K. Aberer, M. Hauswirth, and A. Salehi. Infrastructure for data processing in large-scale interconnected sensor networks. In *MDM'07*, pages 198–205, 2007.
2. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
3. E. Bouillet, M. Feblowitz, Z. Liu, A. Ranganathan, A. Riabov, and F. Ye. A semantics-based middleware for utilizing heterogeneous sensor networks. In *DCOSS'07*, pages 174–188, 2007.
4. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.
5. D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *ISWC'11*, October 2011.
6. J. F. Sequeda and O. Corcho. Linked stream data: A position paper. In *SSN'09*, 2009.
7. A. P. Sheth, C. A. Henson, and S. S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 12(4):78–83, 2008.
8. K. Whitehouse, F. Zhao, and J. Liu. Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data. In *EWSN'06*, pages 5–20, 2006.

Appendix: Meeting the Semantic Web Challenge Requirements

Minimal requirements

√ **The application has to be an end-user application.** Each LSM deployment has a user-friendly web interface for different user levels as showed in Section 2. Other interesting functionalities are also demonstrated at the LSM's user manual at `http://code.google.com/p/deri-lsm/`.

√**The information sources used should be under diverse ownership or control.** By design, LSM allows to collect data from different providers under flexible ownership and control policies. LSM also allows users to annotate and integrate existing data sources to create new data ones, which can have accessing/publishing policies defined by users.

√**The information sources used should be heterogeneous.** As showed in Section 4, LSM supports a variety of data formats and accessing protocols.

√**The information sources used should contain substantial quantities of real world data.** The system's ultimate goal is to make real world data captured by sensors as linkable Web information sources. Our current deployment at `http://lsm.deri.ie/` involves over 100,000 live sensor data sources all over the world with approximately 10-20 million updates per day.

√**The meaning of the data has to play a central role.** Section 3 shows that the meaning of sensor data is represented by ontologies. The user annotation process creates transformation rules to allow wrappers to transform sensor readings into meaningful and integrable data. As data sensor items are represented in linkable data objects and annotated with meaningful links, the unified processing model offered by the query processors allows LSM to achieve the composability of sensor data that traditional technologies from sensor data management can not have.

Additional Desirable Features

√**The application provides an attractive and functional Web interface (for human users).** LSM provides an easy-to-use GUI which has several visualisations of sensor data such as map, charts, animated data updates (see Figure 1 and user manual). The interactive facet search makes the data exploration more intuitive and efficient. The server-push technology brings the experience of real-time web rendering to the users.

√ **The application should be scalable.** The scalability of the system depends on underlying triple storage technologies, e.g. Virtuoso and the Linked Stream Data processing engine (CQELS) [5]. Currently, the static dataset for around 100,000 sensor data sources contains around 20 million triples. We are constantly adding more data sources and enriching more metadata on the assumption that Virtuoso can handle billions of triples. For processing continuous queries, CQELS engine can handle up to 50,000 updates per seconds with thousands of query instances.

√**Novelty, in applying semantic technology to a domain or task that have not been considered before.** Applying semantic technology to sensor data is a new trend in data integration, and this work has been pioneering towards this trend. It is one of the first systems that allows the integration of live sensor data with data in the Linked Data Cloud.

√ **The application has clear commercial potential and/or large existing user base.** This platform provides an easy and unified way to integrate useful data captured from sensors. Its Linked Stream Data can be used to rapidly build interesting applications in different domains like "smart cities", e-health and tourism.

√ **Multimedia documents are used in some way.** Some sensor data sources supported in LSM are images, audios or videos such as traffic camera, satellite/radar images and noise sensors.

√ **There is an use of dynamic data, perhaps in combination with static information.** Dynamic Linked Stream Data combined with static data from Linked Data Cloud is the main feature and motivation of the system.

√ **There is support for accessibility on a range of devices.** LSM publishes data in several formats to simplify the data consumption in different platforms. An example with the Android platform is shown in Figure 3.

Acknowledgement This work has been supported by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-II) and by the Irish Research Council for Science, Engineering and Technology (IRCSET).