Short Paper: From Streaming Data to Linked Data – A Case Study with Bike Sharing Systems

Edna Ruckhaus[1,2], Jean-Paul Calbimonte[2], Raúl García-Castro[2], and Oscar Corcho[2]

[1] Universidad Simón Bolívar, Venezuela
ruckhaus@ldc.usb.ve
[2] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
{jpcalbimonte,rgarcia,ocorcho}@fi.upm.es

Abstract. Current methods and tools that support Linked Data publication have mainly focused so far on static data, without considering the growing amount of streaming data available on the Web. In this paper we describe a case study that involves the publication of static and streaming Linked Data for bike sharing systems and related entities. We describe some of the challenges that we have faced, the solutions that we have explored, the lessons that we have learned, and the opportunities that lie in the future for exploiting Linked Stream Data.

1 Introduction

The process of publication of a dataset as Linked Data is composed of several activities: specification, modeling, generation, publication, and exploitation [1]. Current methods and tools that support Linked Data publication have been designed for static data without considering the growing amount of streaming data available on the Web. Nevertheless, streaming data is being used in a large number of domains (financial, environment, transport, energy, among others), and may be generated by physical sensors, by software systems or even by humans.

The representation of streaming data following the principles of Linked Data facilitates its integration to the diverse datasets in the Linked Data cloud, and also to other private Linked Data datasets. Streaming data usually coexists with static data, either because there is static data associated with the sensors that produce data streams, usually in the form of descriptive information on the sensor platform and observations (e.g., platform name, location, observed property), or because there are links to static data published in another dataset. Additionally, user requirements may be related to statistics on streaming data that have been accumulated over time, so historical streaming data needs to be considered in the different steps of linked stream data publication.

In this paper we report on our experience on exposing as Linked Data some static and dynamic data available in the Spanish cities of León and Zaragoza. More specifically, we have focused on static data about points of interest, available from Government open data portals, and on dynamic data about bicycle sharing systems.

2 Case Study: Bicycle Sharing Systems

Several bicycle sharing systems in cities all over the world have made their data available on the Web. Bike rental stations are distributed in different points in the city and the bike sharing systemusually allows users to pick up a bike at any station and drop it off at any (other) station. The goal of this case study is to publish and exploit up-to-date Linked Data about the availability of bikes and free slots in the stations, and links to related resources like travel guides and points of interest (e.g., museums, restaurants).

At this first stage, we are using the data rendered by the services provided by the CityBikes API[1], focusing on the bike sharing systems from the Spanish cities of León and Zaragoza. Besides, we have connected this streaming data to static data from open data portals from these cities, on museums and libraries (for León), and on restaurants (for Zaragoza), and to data about travel guides from El Viajero [2].

3 Data Publication Activities

In this case study, we followed a process inspired by the method proposed in [1], which envisions a continuous process that consists of five main activities: specification, modelling, generation, publication and exploitation.

3.1 Specification

The data sources selected for this case study were related to resources that could be useful to locals and visitors that use bike sharing systems in the different cities.

The data for bike sharing systems were obtained in JSON through the CityBikes API. Data for restaurants, museums, libraries and guides is published in RDF by open data portals from the cities of León and Zaragoza[2]. While data from Zaragoza was generally of high quality, we found some problems with data from León, namely: invalid URIs, points of interest that were either not geo-located or with a geo-position but no location name, different coordinate systems for geographic positions, and timestamps that did not represent the timezone where they were located. After some interactions, these problems were corrected by the data providers themselves; this shows that data reuse can help in the curation of data sources. In the case of the different coordinate systems, the inconsistency was solved during the RDF generation step.

3.2 Modelling

We have built the citybikes ontology network[3] to represent knowledge related to available bikes and free slots in bike sharing systems. These measurements represent the state of a bike station in a particular place and time and are measured through a sensor in each station. The citybikes ontology network follows a modular structure consisting

[1] `http://api.citybik.es/`
[2] `http://www.datosabiertos.jcyl.es/`, `http://www.zaragoza.es/ciudad/risp/`
[3] `http://transporte.linkeddata.es/files/citybikesontologynetwork.zip`

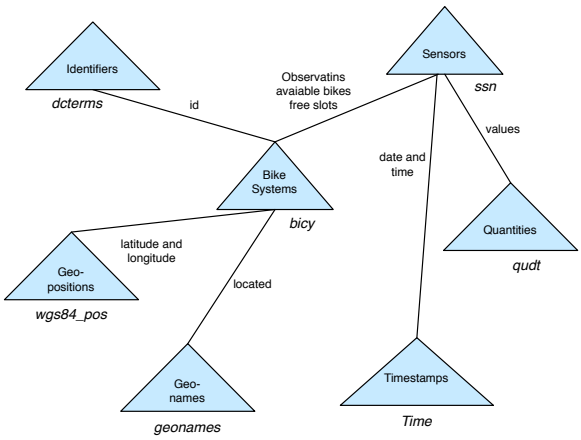

Fig. 1. Bike Sharing Ontologies Network

of a central ontology that is related to a set of ontologies that describe the different sub-domains involved in the modelling of the bike station measurements (Figure 1).

The central ontology is the bike sharing system ontology, which contains concepts such as the bike system and its name, the station and its name, number, internal id, status, description, number of boxes, free slots and free bikes. We reuse the following ontologies: Semantic Sensor Network[4] for sensors and observations, Geonames[5] to define the location of the systems and the stations based on their latitude and longitude, W3C time ontology[6] to represent the timestamp as an instant, Dublin Core[7] for identifiers, WGS_84[8] for geo positioning bike sharing systems and stations with latitude and longitude, and QUDT[9] for the number of available bikes and free slots.

Once the ontology was developed, we defined a resource naming strategy to ensure that every class in the ontology can have individuals with unique identifiers (i.e., URIs). For this, it was necessary to identify the cardinalities of the properties in the ontology, since information on the "conceptual schema" of the data sources was not explicit.

3.3 RDF Data Generation and Publication

In our use case and in sensor applications in general, we require publishing and consuming stored and streaming data. The first type, also referred to as static, is often related to the metadata and contextual information about sensor data, including geographical location, sensor and station characteristics, observed features, and also includes data

[4] http://purl.oclc.org/NET/ssnx/ssn#
[5] http://www.geonames.org/ontology#
[6] http://www.w3.org/2006/time#
[7] http://purl.org/dc/terms/
[8] http://www.w3.org/2003/01/geo/wgs84_pos
[9] http://qudt.org/

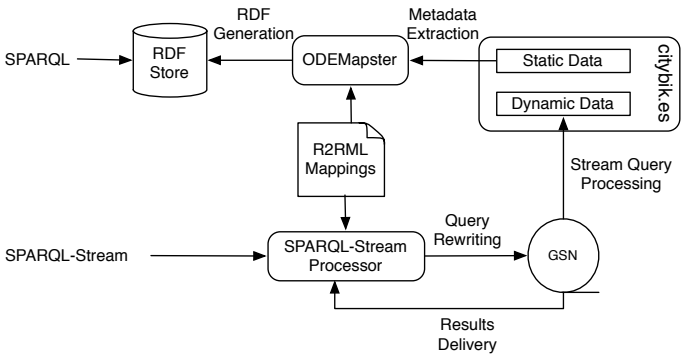

Fig. 2. Approach to generate static RDF and to query RDF streams with SPARQL$_{\text{Stream}}$

about points of interest (museums, libraries, etc.). Streaming data, in contrast, are highly dynamic and are centered on the observations of free slots and available bikes.

For static data, there are methods and tools for generating RDF and publishing them as Linked Data. Some of these tools generate the RDF data using declarative mappings such as R2RML[10] from relational databases, in bulk load operations. This approach is effective for static data, as it is seldom updated. However, for the dynamic sensor observations and for performing continuous queries, this is not the case. The available tools are unsuitable for materializing data in real time, and even if they were, the SPARQL query language does not consider streaming data operators such as time windows.

Therefore, we used a different approach (see Figure 2), based on the idea of reusing streaming sensor data processing engines, that are able to process highly dynamic data efficiently, and using temporal constructs. In order to use such engines, we rely on ontology-based query rewriting of SPARQL queries with streaming extensions. We used one such extension, SPARQL$_{\text{Stream}}$ [3], that is able to use Data Stream Management Systems (DSMS), Complex Event Processors (CEP) and Sensor middleware as underlying query processors. In particular, we used GSN [4] (Global Sensor Networks), a widely used sensor data processor to which we added a wrapper to the CityBikes API. One of the advantages of using SPARQL$_{\text{Stream}}$ is that it uses R2RML mappings to rewrite queries into expressions that can be instantiated and executed by streaming query engines [5]. This allowed us to use the same set of mapping definitions for both static and streaming data. We used ODEMapster[11] for static data and SPARQL$_{\text{Stream}}$ for dynamic data. Notice that both approaches follow very different RDF management strategies: for static data we generate materialized RDF triples that can be later queried in a standard triple store; for dynamic data we pose queries to a virtual streaming RDF dataset, and the queries are rewritten by a SPARQL$_{\text{Stream}}$ processor to the underlying stream processing engine, which throws the query results.

When defining the R2RML mappings it was sometimes necessary to define an object map that has a join condition specifying a child and a parent triples map. As the

[10] W3C RDB2RDF Mapping Language: `http://www.w3.org/TR/r2rml/`
[11] ODEMapster RDB2RDF Tool: `http://neon-toolkit.org/wiki/ODEMapster`

generated RDF property flows from child to parent, it was sometimes necessary to define an inverse property in the ontology only for this purpose. This is the case of the property *isStationOf* which is the inverse of the property *hasStation*.

An interesting requirement in our use case is the possibility of accessing historical data from the dynamic observations. In order to perform data analysis or compare live data with historical records, we cannot directly use the SPARQL$_{\text{Stream}}$ approach as it stands, because it only considers recent observation values. As an alternative, we chose to use the live SPARQL$_{\text{Stream}}$ service in order to periodically pose `CONSTRUCT` queries that generate RDF triples, which can be later imported to the static RDF store. Then, users requiring statistical or analysis queries, can directly use such a store.

The linking activity was completed with the Silk platform [6]. The activity consisted in geo-linking the stations with the points of interest and the travel guides. Silk provides a similarity measure for geo-linkage that requires that both datasets be annotated with the WGS_84 ontology. Some of the data sources, e.g., museums in the city of León, that were considered in this use case were not annotated with this vocabulary so it was necessary to use numeric similarity and aggregated euclidean distance to compute the similarity between two entities. It is not clear if this gives us a precise estimate of the proximity of two points and it was difficult to validate the adequacy of the links.

Care was taken to ensure that the publication technologies satisfy the licensing and access policies previously defined.

3.4 Exploitation

Queries can be posed against static data, dynamic data, and a combination of both. For dynamic data there is no RDF data generation; instead, queries are rewritten by a SPARQL$_{\text{Stream}}$ processor, to the underlying stream processing engine which throws the query results. Our SPARQL$_{\text{Stream}}$ processor can not yet construct in an optimised manner queries that combine streaming and static data, and this is a pending task for which we will adopt some existing strategy [7, 8]. For historical dynamic data, the SPARQL$_{\text{Stream}}$ service is used to periodically pose CONSTRUCT queries that generate RDF. In our case, we have made the data available in two different types of endpoints (one for static data and the other one for dynamic data), and we will soon provide a map-based interface based on the Map4RDF platform[12]. The complete description of the application is available online[13] along with sample queries.

4 Discussion and Open Questions

Some of the main challenges that we have faced in the process of producing and exploiting data in the context of this case study are:

- The SSN ontology has been applied to the domain of bike sharing systems, which falls outside areas where the SSN ontology has been extensively used, such as environmental measurements and agriculture. The ontology has proven to be useful

[12] `http://www.oeg-upm.net/index.php/es/downloads/172-map4rdf`
[13] `http://transporte.linkeddata.es/`

to model the observations and measurements. However, for people not used to the Observation and Measurements approach, which underlies the SSN ontology, this type of modelling is complex to understand. Clear descriptions of usage patterns are needed so that the SSN ontology can be more easily exploited by developers.
- For ontology developers, the SSN ontology still poses some challenges when it has to be extended. For instance, the treatment of properties that are measured by sensors is represented through a class, and when it needs to be extended, it is sometimes unclear how this extension has to be done (e.g., using a subclass or an instance) and it is unclear how properties can be modelled for better reuse across other ontologies and domains.
- There is a need to explore when it makes sense to follow a native RDF stream approach to deal with stream data (e.g., as done in [7, 8]) or an R2RML-based query rewriting approach, as we do here in our work.
- When using the R2RML approach, there are some limitations in the treatment of direct and inverse properties and how they can be defined in the R2RML mappings. This forces ontology developers to overspecify some properties, by defining properties that are inverse to others that are defined. While this does not really represent a major problem from an ontology development perspective, it is important to give clear guidelines to developers in this respect.

Acknowledgements

This work has been supported by the Ciudad2020 INNPRONTA project (IPT-20111006). We want to thank Boris Villazón-Terrazas and Freddy Priyatna for their help during the RDF generation process.

References

1. Villazón-Terrazas, B., Vilches-Blázquez, L., Corcho, O., Gómez-Pérez, A.: Methodological guidelines for publishing government linked data linking government data. In Wood, D., ed.: Linking Government Data. Springer New York (2011) 27–49
2. Garijo, D., Villazón-Terrazas, B., Corcho, O.: A provenance-aware linked data application for trip management and organization. In: 7th Int. Conference on Semantic Systems. (2011)
3. Calbimonte, J.P., Corcho, O., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: Proc. 9th International Semantic Web Conference. (2010) 96–111
4. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: 32nd International Conference on Very Large Databases. (2006) 1199–1202
5. Calbimonte, J.P., Jeung, H., Corcho, O., Aberer, K.: Enabling query technologies for the semantic sensor web. Int. J. on Semantic Web and Information Systems (to appear) **8** (2012)
6. Julius, V., Christian, B., Martin, G., Georgi, K.: Discovering and maintaining links on the web of data. In: ISWC2009. (2009) 650–665
7. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: C-SPARQL: a continuous query language for RDF data streams. Int. J. Semantic Computing **4**(1) (2010) 3–25
8. Le-Phuoc, D., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: ISWC2010. (2011)