SensorMasher : publishing and building mashup of sensor data

Danh Le Phuoc
(Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland
danh.lephuoc@deri.org)

Abstract: SensorMasher is a platform which makes sensor data available following the linked open data principle and enables the seamless integration of such data into mashups. SensorMasher publishes sensor data as Web data sources which can then easily be integrated with other (linked) data sources and sensor data. Raw sensor readings and sensors can be semantically described and annotated by the user. These descriptions can then be exploited in mashups and in linked open data scenarios and enable the discovery and integration of sensors and sensor data at large scale. The user-generated mashups of sensor data and linked open data can in turn be published as linked open data sources and be used by others.

Keywords: sensor web, sensor mashup, linked data
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1

1 Introduction

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, analyzed and interpreted.*`` Making sensor-generated information usable as a new and key source of knowledge will require its integration into the existing information space of the Web.

Sensed data is often archived or streamed as raw data, but rarely associated with enough metadata describing its meaning. Meaning of sensor data includes the feature of interest, the specification of measuring devices, accuracy, measuring condition, scenario of measurements, location, etc. Such metadata is essential when the user is confronted with large numbers of sensors and gigabytes of sensor data. Especially, when the user does not have clear ideas about what he/she is looking for, he/she can start a quite general search of relevant concepts and narrow it down based on semantic descriptions and their relations. For example, a city planner may want to assess and monitor quality of life in certain area. To do so, he/she can start to navigate from his/her own domain of knowledge,then finds out that quality of life depends on noise, sunlight, humidity, air pollution, traffic condition, etc. Then he can filter out which sensor sources can provide such data in his area of interest.

To enable easy access to sensor data to non-technical users, we propose our SensorMashup platform as a step towards the vision of ``The Web of Thing". Our platform will enable the publishing of sensors and the associated data sources as Web citizens under URIs. These published entities, called Sensor Mashups, will be created and then linked to ontological concepts, other (virtual) sensors and other Web citizens under URIs through a visual composer. This phase will create useful linked data for sensor discovery. Hence, the visual composer is able to provide an intuitive GUI which allows users to navigate and explore sensor data sources by following semantic links and using faceted-browsing techniques. After having found the relevant sensor data sources, the user is able to combine them visually in a workflow editor and then connect them to data processing operators to create a new Sensor Mashup. Sensor data published from our platform can be accessed through SPARQL endpoints and RESTful web services under JSON, XML, RDF formats which can be easily used by various applications. Triple-based query support enables semantic-based traversal over sensor data, thus makes the data filtering and correlating more sophisticated than those of previous approaches. Triple-based query support is only used to enable access via standard mechanisms while internally we use compressed data representations for efficiency reasons.

2 SensorMasher

In order to link real world data captured by sensors to the Web, it has to become a form of Web resource. Sensors and their data elements can become Web resources by publishing their data on the Web using linked data techniques. When adopting such an approach, sensors and their data are assigned a URI which enables the creation of links to other Web entities. Each sensor has a URL pattern for streaming real-time data as well as historical data. This pattern also includes semantic descriptions of the sensor and sensor data model as described in the following.

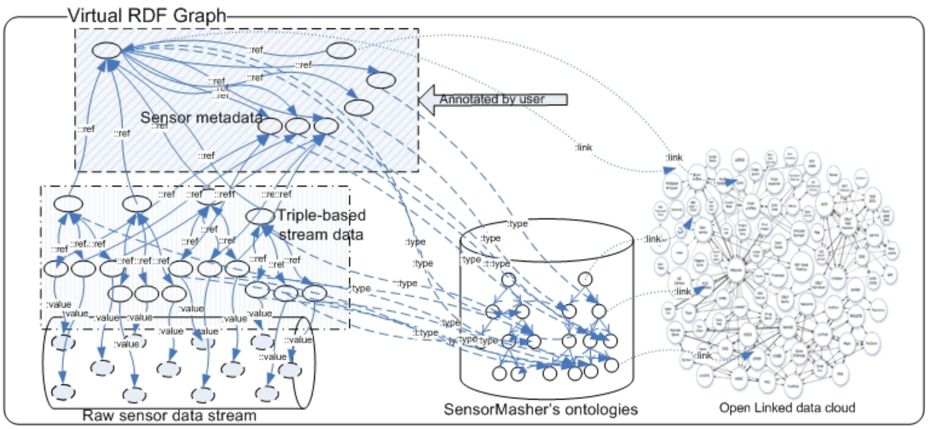


Figure 1: Virtual RDF graph of sensor data stream

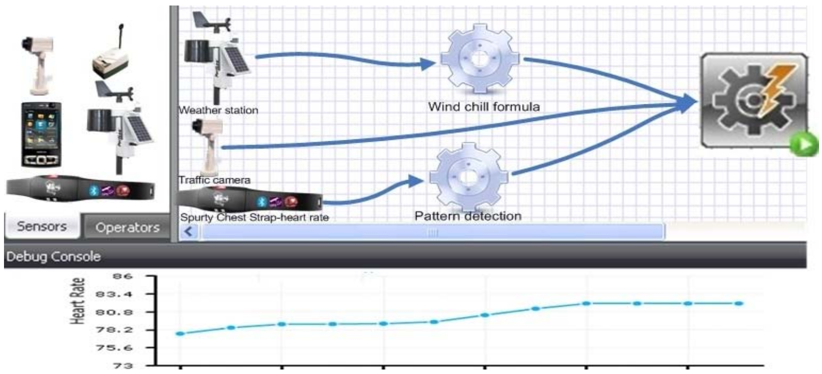
As sensors and their data elements have URI identifiers, we are able to link them into a virtual RDF graph as shown in Figure 1. The graph is called a *virtual graph*, because the whole graph is not materialized and stored in a triple storage. It is constituted by interlinked subgraphs controlled by sensor mashups. A mashup's metadata stored in the metadata repository defines how the raw sensor data in a data stream can be linked to domain knowledge and external linked data. This is driven by a set of ontologies which are used to capture facts and the data model of the sensor information system. This linked metadata will guide the finding of triple patterns needed from the virtual RDF graph of sensor data.

The system is accessible online at *sensormasher.deri.org* (source code is available at *sourceforge.net/projects/sensormasher*). As sensor data coming to the system as data stream under relation, we extended SQL to support declarative continuous queries over stream data. These declarative queries will be translated into physical query plans over the underlying relational data storage. Because there are only some minor modifications compared to SQL, querying data streams with this DSMS is basically similar to querying relations. Hence, following the D2RQ[Bizer,06] approach we extended Jena ARQ[Jena] to build the SPARQL query processor on top of the Data Stream Management System and the metadata repository. The mapping rules, which are similar to the D2RQ mapping language, are automatically generated from the configurations of the mashups.

We use two main ontologies to control the SensorMasher: the core ontology and the extended ontology. The core ontology represents core concepts and properties which are the same for every SensorMasher deployment. The extended ontology contains subclasses of the core ontology which can be customized based on specific requirements. In our prototype, the extended ontology was built by using concepts and properties from the Physical Property taxonomy in the SWEET Property ontology[Sweet] and the SANY Sensor Taxonomy[Taxonomy].To support implicit properties inferred from ontological relationships such as subClassOf, subPropertyOf, inverseOf, etc. in answering SPARQL queries, we use the Jena in-memory reasoner to reason about these two ontologies at class level to generate query mapping rules. For example, if we query for sensors that can measure meteorological measurement, the result will contain sensors that can measure temperature, humidity, wind speed, etc.

3 Demonstrations

We would like to demonstrate the Mashup Composer by which user is able to visually drag and drop sensors as data sources and connect them to data processing blocks to create new sensor data sources as depicted in figure 2.



Firgure 2: Sensor Mashup Composer

Via Mashup&Sensor Explorer (depicted in figure 3), user is able to filter and explore sensor data sources by drawing a polygon on the map to locate the area of interest. The user is also able to drill down into the result set by using facet filters on types and properties. Furthermore, from an result item, it is possible to follow its relationships to discover other items which have semantic links to the previous one.

Figure 3: Mashup and Sensor Explorer

After showing how to explore and compose sensor mashup, we also show how sensor, mashup description and RDF output of sensor data looks like as depicted in figure 4. Furthermore, the sensor data also can be retrieved in other formats such as RDF, JSON, XML, RSS, CSV,etc.

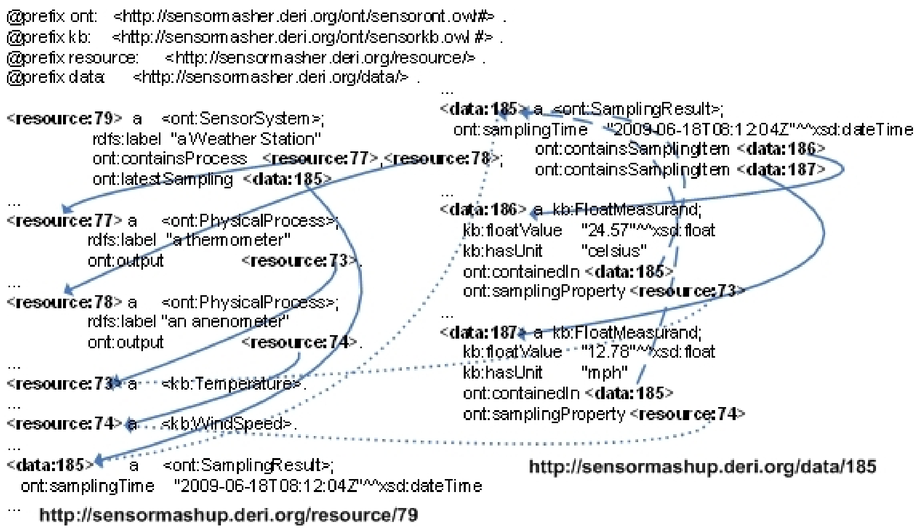


Figure 4 : Sensor description and data in RDF

As we provides SPARQL endpoint to query sensor all types of sensor data, we aslo show how to query over real-time sensor data "the temperature and location where the wind speed has reached 30 miles per hour around the Cliff of Moher" (depicted in figure 5). With this SPARQL endpoint, we also demonstrate the spatial and temporal queries over sensor data.

```
@prefix ont:   <http://sensormasher.deri.org/ont/sensoront.owl#> .
@prefix kb:    <http://sensormasher.deri.org/ont/sensorkb.owl#> .
@prefix spatial:    <http://sensormasher.deri.org/ont/spatialont.owl#> .

SELECT  ?tempReading, ?lat, ?long
WHERE {
?sensorSys rdf:type ont:SensorSystem . ?sensorSys ont:locatedAt ?aFixLoc.
?aFixLoc rdf:type  spatial:FixedLocation.?aFixLoc spatial:lat ?lat. ?aFixLoc spatial:long ?long.
?aFixLoc  spatial:within ?area. ?area  spatial:geometryObjName "Cliff of Moher".
?sensorSys ont:output ?temperature. ?temperature rdf:type kb:Temperature.
?sensorSys ont:output ?winspeed. ?windspeed rdf:type kb:Windspeed.
?sensorSys ont:latestSampling ?sampling.
?tempReading ont:containedIn ?sampling. ?tempReading ont:samplingProperty ?temperature.
?wsReading ont:containedIn ?sampling. ?wsReading ont:samplingProperty ?windspeed.
FILTER(?wsReading kb:floatValue>30)
}
```

Figure 5 : Query sensor data via sparql endpoint

References

[Bizer, 06] Bizer, C., Cyganiak, R.: D2R Server Publishing Relational Databases on the Semantic Web. In: ISWC'06.(2006)9.

[Jena] Jena Semantic Web Framework, http://jena.sourceforge.net/

[Sweet] Semantic Web for Earth and Environmental Terminology (SWEET) 1.1, Physical Property. http://sweet.jpl.nasa.gov/1.1/property.owl

[Taxonomy] Sensor Taxonomy. http://sany-ip.eu/publications/1954