Linked Sensor Data

Harshal Kamlesh Patni
Wright State University - Main Campus

Cory Andrew Henson
Wright State University - Main Campus

Amit P. Sheth
Wright State University - Main Campus, amit.sheth@wright.edu

Follow this and additional works at: http://corescholar.libraries.wright.edu/knoesis
Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation
http://corescholar.libraries.wright.edu/knoesis/545

This Conference Proceeding is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
Linked Sensor Data

Harshal Patni, Cory Henson, Amit Sheth
Kno.e.sis Center, Department of Computer Science and Engineering,
Wright State University, Dayton, Ohio, USA
{harshal, cory, amit}@knoesis.org

ABSTRACT

A number of government, corporate, and academic organizations are collecting enormous amounts of data provided by environmental sensors. However, this data is too often locked within organizations and underutilized by the greater community. In this paper, we present a framework to make this sensor data openly accessible by publishing it on the Linked Open Data (LOD) Cloud. This is accomplished by converting raw sensor observations to RDF and linking with other datasets on LOD. With such a framework, organizations can make large amounts of sensor data openly accessible, thus allowing greater opportunity for utilization and analysis.

KEYWORDS: Linked Data, Semantic Sensor Web, Sensor Data, Sensor Web Enablement, Dataset Generation, Resource Description Framework (RDF)

1. INTRODUCTION

In 2008, Hurricane Ike, the largest hurricane ever observed in the Atlantic basin and the third most destructive hurricane on record, made its landfall in the United States. During this time, a number of weather organizations collected data from thousands of sensors deployed in the United States. Such data can be used to answer aggregate queries spanning both temporal and geographical areas. While these queries may seem easy to answer, computing the solutions using traditional storage formats is non-trivial.

Let us consider the following scenario. You are interested in calculating the average WindSpeed when hurricane Ike was active in Texas. To accomplish this task, we need to know the time for which the hurricane was active in Texas, the sensors deployed in Texas, and the WindSpeed measurements from these sensors. However, the data is only accessible to the organization involved in the collection of such data. Such data is very useful for research and analysis, and having access to such sensor datasets helps in benchmarking, standardization, and comparison of sensor related technologies. The main idea of this paper is to make sensor data and metadata publically accessible by storing it in the Linked Open Data Cloud [1]. Following the trend of making data publicly accessible, Linked Sensor Data is an effort to add sensors domain to the LOD [1]. In contrast to majority of the Linked Open datasets Linked Sensor Data has been generated using a Sensor Ontology [8] as its schema.

With a view of addressing this issue, we discuss background information in section 2. In section 3, we describe two sensor datasets that we have created and made accessible on LOD. Section 4 gives detailed description on dataset generation and section 5 gives information on an API, used to convert Observations and Measurements (O&M), a standard language for encoding sensor observations, to RDF. Section 6 discusses issues with the current conversion tools. Section 7 describes a prototype application built over the Linked Sensor Data. Section 8 gives related work and the paper concludes in section 9. Key contributions of this paper include:

- Generating RDF datasets for sensor descriptions and observations.
- Making sensor datasets available on LOD to allow querying and analysis over collected sensor descriptions and observations.
- Creating an API for converting O&M to RDF, enabling other organizations to provide their sensor datasets on LOD

2. BACKGROUND
Linked Open Data Cloud - The goal of Linked Data is to enable people and organizations to share structured data on the Web as easily as they can share documents today. The term Linked Data was coined by Tim Berners-Lee in his Linked Data Web architecture note. [2] Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF". [1] URIs are used to identify any kind of object or concept on the LOD. Resource Description Framework (RDF) is a general-purpose language for data representation on the Web. [3] The basic tenets of Linked Data are described as follows:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. So that they can discover more things.[2]

Linked Data is a large and growing collection of interlinked public datasets encoded in RDF that span diverse areas such as: life sciences, nature, science, geography and entertainment. In the sensors domain, sources of geospatial information such as GeoNames and LinkedGeoData [4] are of particular importance. The GeoNames geographical dataset contains over eight million geographical names and consists of 7 million unique features including 2.6 million populated places and 2.8 million alternate names. [5] In section 3, we will introduce two new sensor datasets, LinkedSensorData and LinkedObservationData, with links to locations defined in GeoNames.

Observation and Measurements (O&M) - Observation and Measurements [6] is one of the OGC[3] Sensor Web Enablement (SWE)3 suite of standards that defines an abstract model and an XML schema encoding for sensor observations. O&M is a widely and commonly accepted standard for encoding sensor observation within the sensors community. In section 5, we demonstrate an API that converts O&M to RDF.

Ontology Model of Sensor Data - In computer science and information science, ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. [7] Our sensors ontology uses the concepts within the O&M standard to define sensor observations. Within the O&M standard, an observation (om:Observation) is defined as an act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property, and a feature (om:Feature) is defined as an abstraction of real world phenomenon. (Note: om is used as a prefix for Observations and Measurements). The major properties of an observation include feature of interest (om:featureOfInterest), observed property (om:observedProperty), sampling time (om:samplingTime), result (om:result), and procedure (om:procedure). Often these properties can be complex entities that may be defined in an external document. For example, om:FeatureOfInterest could refer to any real-world entity such as a coverage region, vehicle, or weather-storm, and om:Procedure often refers to a sensor or system of sensors defined within a SensorML 6 document. Therefore, these properties are better described as relationships of an observation. The Sensor Data ontology can be found at [8].

Resource Description Framework (RDF) - The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) 7 specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources using a variety of syntax formats. [9] RDF extends the linking structure of the Web to use URIs for naming relationships between things as well as the resources that are related (usually referred to as a “triple”). Using this simple model allows structured and semi-structured data to be mixed, exposed and shared across different applications. This linking structure forms a directed, labeled graph where the edges represent the named link between two resources represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. [10]

SPARQL Protocol and RDF Query Language (SPARQL) - As mentioned above, RDF is a directed, labeled graph data format for representing information in the Web. SPARQL [11] is a syntactically-SQL-like language for querying RDF graphs. [12]

3. SENSOR DATASET DESCRIPTION

---

1 http://www.w3.org/People/Berners-Lee/
2 http://www.wikipedia.org/
3 http://www.opengeospatial.org/
4 http://www.opengeospatial.org/ogc/markets-technologies/swe
5 www.w3.org/XML/Schema
6 http://www.opengeospatial.org/standards/sensorml
7 http://www.w3.org/
With a view of addressing the above issue of making sensor data publicly accessible on the LOD, we have generated two datasets, LinkedSensorData and LinkedObservationData.

**Linked Sensor Data** - LinkedSensorData is an RDF dataset containing expressive descriptions of ~20,000 weather stations in the United States. The data originated at MesoWest, a project within the Department of Meteorology at the University of Utah that has been aggregating weather data since 2002. [13] On average, there are five sensors per weather station measuring phenomena such as temperature, visibility, precipitation, pressure, wind speed, humidity, etc. In addition to location attributes such as latitude, longitude, and elevation, there are links to locations in Geonames [3] near the weather station. The distance from the Geonames location to the weather station is also provided. The dataset also contains links to the most current observation for each weather station provided by MesoWest [13]. This sensors description dataset is now part of the LOD.

**Linked Observation Data** - LinkedObservationData is an RDF dataset containing expressive descriptions of hurricane and blizzard observations in the United States. The data again originated at MesoWest. [13] The observations collected include measurements of phenomena such as temperature, visibility, precipitation, pressure, wind speed, humidity, etc. The weather station’s observations also include the unit of measurement for each of these phenomena as well as the time instant at which the measurements were taken. The dataset includes observations within the entire United States during the time periods that several major storms were active -- including Hurricane Katrina, Ike, Bill, Bertha, Wilma, Charley, Gustav, and a major blizzard in Nevada in 2002. These observations are generated by weather stations described in the LinkedSensorData dataset introduced above. Currently, this dataset contains more than a billion triples. The RDF dataset for each of the above storms is available for download in gzip format at [14]. The statistics for these storms can be found in the Table 1 below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Storm Type</th>
<th>Date</th>
<th>No Of Triples</th>
<th>No Of Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td></td>
<td></td>
<td>1,735,284,759</td>
<td>159,490,500</td>
</tr>
<tr>
<td>Bill</td>
<td>Hurricane</td>
<td>August 17 - August 22, 2009</td>
<td>231,251,218</td>
<td>21,737,709</td>
</tr>
<tr>
<td>Ike</td>
<td>Hurricane</td>
<td>September 1 - September 13, 2008</td>
<td>374,094,060</td>
<td>34,430,964</td>
</tr>
<tr>
<td>Gustav</td>
<td>Hurricane</td>
<td>September 1 - September 13, 2008</td>
<td>258,378,511</td>
<td>23,792,818</td>
</tr>
<tr>
<td>Bertha</td>
<td>Hurricane</td>
<td>July 6 - July 17, 2008</td>
<td>278,235,724</td>
<td>25,762,568</td>
</tr>
<tr>
<td>Wilma</td>
<td>Hurricane</td>
<td>October 17 - October 23, 2005</td>
<td>171,856,406</td>
<td>15,797,032</td>
</tr>
<tr>
<td>Katrina</td>
<td></td>
<td>August 23 - August 30, 2005</td>
<td>201,286,049</td>
<td>18,832,041</td>
</tr>
<tr>
<td>Charley</td>
<td>Hurricane</td>
<td>August 9 - August 15, 2004</td>
<td>101,956,700</td>
<td>9,333,676</td>
</tr>
<tr>
<td>Blizzards</td>
<td></td>
<td>April 1 - April 6, 2005</td>
<td>117,352,227</td>
<td>10,327,792</td>
</tr>
</tbody>
</table>

Table 1. Statistics for LinkedObservationData Dataset

**Access to Datasets on LOD** - Both the datasets, LinkedSensorData and LinkedObservationData are hosted on Virtuoso RDF. Virtuoso RDF is an open source triple store provided by OpenLink Software. [15] Virtuoso RDF provides a SPARQL endpoint to query these datasets which can be found at [16]. More information about querying the dataset can be found at [14]. However, for users who don’t know SPARQL, we connected Pubby® (a browser interface for accessing Linked Data) over Virtuoso RDF so that the triples in the store can be viewed in an intuitive style. The Pubby interface can be found at [17].

**4. SENSOR DATASET GENERATION**

The data generation workflow is comprised of four main parts, as shown in Figure 1. The workflow begins with MesoWest’s Service that provides sensor observations encoded as comma separated numerical values representing the intensities of phenomena measured by sensors. The sensor observations are then converted to O&M. Since the main idea of this paper is to make the sensor descriptions and observations available on the Linked Open Data Cloud, the O&M first needs to be converted to RDF. The RDF generated is then stored in a Virtuoso RDF knowledgebase. Since O&M has XML syntax, we attempted to reuse several open source XML2RDF tools in order to convert O&M to RDF. However, we found several issues with this approach that will be discussed in Section 6. Due to this failed attempt in using open source XML2RDF conversion tools and since much sensor data is encoded in O&M, we thought it useful to write an API that converts O&M to RDF, discussed in section 5.

**Phase 1** - The first phase is comprised of querying MesoWest for observational data and parsing the result. MesoWest provides a service to access past data that takes station ID, date, and time as parameters. It returns an HTML page with the observational values for different sensors embedded within the weather station for a period of 12 hours ending at the date and hour provided. Since we already know when the hurricanes occurred, we are

8 http://www4.wiwiss.fu-berlin.de/pubby/
able to query this service for all the sensors in the US with the dates for which the hurricanes were active. The resulting HTML page is then parsed to extract the sensor observation data.

**Phase 2** - The second phase consists of converting the raw textual data retrieved from MesoWest into O&M. As described in the phase 1, the sensor observations parsed from the html page were fed to an XML parser. We used the SAX (Simple API for XML) parser to generate the O&M. Figure 2 encodes the location—latitude, longitude, and elevation—of the station, ID “AR030” in O&M.

![Fig.2. Location Attributes encoded in O&M](image)

Figure 3 encodes the phenomena; the station is capable of measuring in O&M. We use Xlink:href to annotate the O&M using the concepts in our ontology as shown in Figure 3. Here href is an attribute in XML markup language that is used for describing links between resources in XML documents. [18] Semantically annotating the concepts in O&M documents using our ontology provides additional metadata.

![Fig.3. Phenomena encoded in O&M](image)

Figure 4 encodes the time for which the station was active in O&M.

![Fig.4. Sampling Time for Sensor Measurement in O&M](image)

Sensor observations for every time instant are encoded as comma separated values in O&M which forms a block and each block is separated by ‘@@’ as shown in Figure 5.

![Fig.5. Sensor Observations for every Time Instant](image)

Figure 6 shows how O&M encodes unit of measurement for a phenomenon.

![Fig.6. Unit of Measurement for AirTemperature in O&M](image)

The O&M generated in this phase forms the input for our **O&M2RDF-Converter** API discussed in section 5. A complete O&M for station ID “AR030” can be found in figures 14 and 15 under the appendix section.

**Phase 3** - The third phase consists of converting sensor observations encoded in O&M to RDF. Since both O&M and RDF have an XML like syntax, we generated an XSLT to convert O&M to RDF. XSLT is basically a language for transforming XML documents into other XML documents [19]. The XSLT forms the main component for our **O&M2RDF-Converter** API. Figure 7 shows the sensor coordinates encoded in RDF (The examples used in this section are RDF-N3 format for ease of explanation.). The corresponding O&M can be found in Figure 2. These figures show the result of converting O&M to RDF using XSLT.

![Fig.7. Location attributes encoded in RDF.](image)

Figure 8 encodes the phenomena; the station is capable of measuring in RDF. The corresponding O&M can be found in Figure 3.

![Fig.8. Phenomena encoded in RDF](image)

---

9 http://www.saxproject.org/
Figure 9 encodes the time for which the station was active in RDF. The corresponding O&M can be found in Figure 4.

```
sens:obs:Instant_2005_08_22_00_05_00
  a owl:time:Instant;
  owl:inXSDDateTime "2005-08-22T00:05:00".
sens:obs:Instant_2005_08_22_23_45_00
  a owl:time:Instant;
  owl:inXSDDateTime "2005-08-22T23:45:00".
```

**Fig. 9.** Sampling Time for Sensor Measurement in RDF

Figure 5 gives a list of generated observations for each of the phenomena the sensor is capable of measuring. The corresponding RDF for a single phenomena measurement can be found in figure 10.

```
sens:obs:MeasureData_AirTemperature_AR030_2005_08_22_23_45_00
  a owl:MeasureData;
  owl:hasObservation sens:obs:Observation_AirTemperature_AR030_2005_08_22_23_45_00.
sens:obs:Observation_AirTemperature_AR030_2005_08_22_23_45_00
  a sens:Observation;
  sens:hasProvider sens:obs:System_AR030;
  sens:hasObservationAirTemperature AR030_2005_08_22_23_45_00.
sens:obs:System_AR030
  a sens:ObservationSystem;
  sens:hasName sens:obs:Component_AirTemperature_AR030.
sens:obs:Component_AirTemperature_AR030
  a sens:Component;
  sens:hasLocation Geonames location.
sens:obs:DataType_AirTemperature
  a sens:DataType.
```

**Fig. 10.** Sensor Observation in RDF

Figure 6 gives Unit of Measurement for a phenomena encoded in O&M. The corresponding RDF can be found in figure 11.

```
sens:obs:MeasureData_AirTemperature_AR030_2005_08_22_23_45_00
  a sens:MeasureData;
  sens:hasUnitValue "84.0";
  sens:hasUOM weather:faahrenheit.
```

**Fig. 11.** Unit of Measurement for AirTemperature in RDF

We added a "hasLocation" property to link our dataset to Geonames dataset on LOD. We also provide a "distance" property that specifies the distance between the weather station and the Geonames location. The RDF generated in this phase is the output of our O&M2RDF-Converter API discussed in section 5. A complete RDF for station ID "AR030" can be found in figures 16 and 17 under the appendix section.

**Phase 4** - The fourth phase consists of storing the RDF in Virtuoso RDF store. OpenLink Virtuoso stores RDF data in repositories. We created one repository for each storm with the repository labeled with the name of the storm. We are currently loading data on Virtuoso RDF store.

**5. O&M2RDF-CONVERTER API**

An API (Application Programmer Interface) is an abstraction that defines and describes an interface for the interaction with a set of functionalities exposed by a component of a software system. The software that provides the functionalities described by an API is said to be an implementation of the API. [20] Our API or Web service, named **O&M2-RDF-Converter**, only accepts POST requests. The server recognizes the POST request and reads the data (O&M as a string) sent from the client. For a Java program to interact with a server-side process it simply must be able to write to a URL, thus providing data to the server. This can be done through the following steps:

1. Create a URL. To access the service, the URL should be [http://knoesis1.wright.edu/stream-servlet/Servlet](http://knoesis1.wright.edu/stream-servlet/Servlet)
2. Retrieve the URLConnection object.
3. Set output capability on the URLConnection.
4. Open a connection to the resource.
5. Get an output stream from the connection.
6. Read the O&M from the file as a string and write the O&M string to the output stream.
7. Close the output stream.

The servlet reads the information sent by the client, converts it to RDF, and then sends the RDF back as a string. You now need to read the string the server has sent back and store it onto a file with RDF extension (.rdf). Sample request and response can be found in figures 18 and 19 respectively under the appendix section.

**6. ISSUES WITH CURRENT TOOLS**

Use of existing tools promotes software reuse, and therefore we attempted to reuse several existing tools for converting O&M to RDF, such as JXML2OWL [21] and XML2OWL [10]. JXML2OWL project was developed in 2006 at Madeira University as a part of the SEED Project. The project is divided in 2 parts, as follows:

- **JXML2OWL API** - an open source library for mapping XML schemas to OWL ontologies on the JAVA platform.
- **JXML2OWL Mapper** - an easy to use standalone application with a graphical user interface developed in JAVA SWING, using the JXML2OWL API.

The JXML2OWL API requires the user to specify the mappings from XML elements to ontology classes and properties. Then it creates an XSLT stylesheet that transforms instances of an XML schema into instances of an existing OWL ontology. However we found a few problematic issues with this tool.

- As shown in Figure 3 the XML element "swe:component" encodes the phenomena that the weather station measures as an attribute. The count of

---

XML element "swe:component" depends on the number of phenomena that the weather station is capable of measuring. However we are interested in only the weather related phenomena concepts and not all the phenomena concepts encoded in O&M. Since JXML2OWL requires the exact mapping between XML element and the OWL class we would need to use XPATH predicates to exactly specify the XML element that encodes the weather related concept we need to map. However JXML2OWL does not allow XPATH predicates and therefore it is difficult to achieve the above.

- As shown in Figure 9 (a subset of our ontology viewed in protégé) we have a class called Location which has a sub-class called Point. The concept Point has properties – latitude, longitude and elevation. JXML2OWL only allows mapping to the topmost class in the hierarchy, which in this case is Location, and not the sub-class, which in this case is Point. It is important for us to map to the Point class since a sensor is located at a point, which has coordinates. Hence, with this model, it is incorrect to map a sensor to a location.

The other tool we looked at was XML2OWL. XML2OWL is an ANTLR\textsuperscript{11}-based program written in C++ for automatic conversion of an XML file to OWL file. The XML2OWL tool takes at least two parameters as input, a rules definition file (.rules) that defines the XML to OWL mappings and an XML instance file. This tool allows XPATH predicates, which was a major problem in using JXML2OWL. XML2OWL tool could serve our purpose, however it is not available as an installable software tool and does not provide a Web Service or an API.

7. APPLICATION ON LINKED SENSOR DATA

The sensor description provided by MesoWest contains geographic coordinates for a weather station. Since Geonames contains geographical names, linking our LinkedSensorData dataset to Geonames gives an advantage of answering sensor discovery queries such as “Find all sensors near Dayton-Wright Brothers Airport”. This query can be found in figure 13.

```sparql
SELECT ?sensor
WHERE
{
 ?sensor om-owl:hasLocatedNearRel ?rel .
}
```

Fig.13. Sample Sensor Discovery Query

We have built an application that provides an easy-to-use map-based GUI in order to discover sensors near a named location. The user provides a location name in the search box and clicks submit. The application will build and execute a SPARQL query over the LinkedSensorData dataset on LOD and displays all the sensors near the given location. The sensors displayed on the map also provide additional information such as phenomena measured and geographic coordinates. In addition, the sensor descriptions contain links to original data sources, such as location information on Geonames and current sensor data on MesoWest. A screenshot of the application can be found in Figure 14.

\textsuperscript{11} http://www.antlr.org/
We have generated datasets of rich sensor description and sensor observation that contains over 1 billion triples. We are currently loading this data onto the Virtuoso RDF store. Once loaded this will be one of the largest datasets on LOD and the first to describe sensor data. The automatic conversion of sensor data to RDF can help many different organizations to make available large amounts of inaccessible and underutilized sensor data. In the future we hope to extend these sensor datasets with relationships to other datasets on LOD, such as LinkedGeoData [4] and DBpedia [24]. DBpedia knowledge base contains concepts within the sensors domain like storms and hurricanes. However DBpedia only gives general information about the storm. Linking DBpedia to LinkedObservationData and/or LinkedSensorData dataset would provide a link between the sensor data and the textual description of the storm.

ACKNOWLEDGEMENT

This research was supported in part by The Dayton Area Graduate Studies Institute (DAGSI), AFRL/DAGSI Research Topic SN08-8: "Architectures for Secure Semantic Sensor Networks for Multi-Layered Sensing."

REFERENCES

5. GeoNames, http://www.geonames.org/
6. Observation and Measurements (O&M), http://www.opengeospatial.org/standards/om
11. SPARQL Query Language, http://www.w3.org/TR/rdf-sparql-query/
17. Pubby for our dataset, http://knoesis.wright.edu/ssw/
19. XSLT, http://www.w3.org/TR/xslt
APPENDIX

8.
24. DBpedia, http://dbpedia.org/About

Fig.14. Sensor Information encoded in Observation and Measurements

Fig.15. Sensor Observations encoded in Observation and Measurements

Fig.16. Sensor Description encoded in RDF N3 syntax
Fig. 17. Sample Sensor Observations encoded in RDF-N3

Fig. 18. Snippet Of Sample O&M request

Fig. 19. Snippet Of Sample RDF response