Undefined 0 (2014) 1–0
IOS Press


# Global Weather Sensor Dataset

Hoan Nguyen Mau Quoc [a,*], Hung Ngo Quoc [a], Manfred Hauswirth [a,b] and Danh Le Phuoc [a]
[a] *Insight Centre of Data Analytics, National University of Ireland, Galway, Ireland*

[b] *Technical University of Berlin, Berlin, Germany*

**Abstract.** The National Oceanic and Atmospheric Administration has recently published their database to address a pressing need for an integrated global database of hourly land surface climatological data. The database is quickly becoming one of the biggest weather dataset which contains approximately 350 gigabytes for over 20.000 weather sensor stations over the world, with observation data from as early as 1900 to present. In this article, we will describe how NOAA dataset can be transformed and published as linked data (Linked NOAA dataset with over 177 billion triples), with the target of making these data publicly accessible and linkable with other linked data sources, by applying Linked Data principles, accessing and processing methods.

Keywords: Linked NOAA Dataset, Weather Sensor Networks, GoT Ontology, Historical Dataset

## 1. Introduction

Recent reports of global climate change and related issues, along with its unpredictable effects to natural live, has created a sense of public urgency for more detailed information on global weather change. Hence, reliable historical climate datasets are needed for accurate trends and variability analysis, ultimately for detection and attribution of climate change. With the rapidly development of sensor technologies, the recording of such dataset changes from human activities to automatically recording by sensors. The National Oceanic and Atmospheric Administration[1] (NOAA) is a scientific agency within the United States Department of Commerce focused on the conditions of the oceans and the atmosphere. NOAA roles a supplier of environmental information products, a provider environmental stewardship services, and a leader in applied scientific research about environment issues and ecosystems. NOAA's raw weather data is published by The NOAA's National Climatic Data Center[2] (NCDC) on the their website[3]. The dataset collects all of the NCDC and Navy surface hourly data (TD3280), NCDC hourly precipitation data (TD3240), and Air Force Datsav3 surface hourly data (TD9956), into one global dataset. This dataset is one of the biggest weather dataset which contains approximately 350 gigabytes for nearly 20.000 weather sensor stations over the world, with observation data from 1901 to present. It includes all observed elements at hourly intervals, such as temperature, dew point, visibility, wind speed and direction, wind gust, cloud data, precipitation, snow depth, and many others.

The NOAA sources are made publicly available and they play an important role in various weather services over the world. However, this dataset are not easy to access and process due to several difficult technical problems, such as resource constraints and access protocols. Moreover, such dataset sometime requires significant knowledge on the concrete deployments, decoding and transforming activities in order

*Corresponding author. E-mail: hoan.quoc@insight-centre.org
[1] `http://www.noaa.gov/`

[2] `http://www.ncdc.noaa.gov/`
[3] `http://www.ncdc.noaa.gov/data-access/quick-links`

to make it human readable. To remedy these problems and draw on the advantages of Linked Data, we try to make these data publicly accessible, by applying well-established Web principles, accessing and processing methods, thus shielding users and climate analytics group from the underlying complexities. Hence, creating a linked dataset for these data sources will provide extremely useful links to the Linked Data Cloud.

This paper presents the description of Linked NOAA Dataset as well as the process of generating it. We will firstly discuss about the NOAA dataset and its format in Section 2. Next, we propose the data modelling for the global weather sensor dataset based on the SSN ontology in Section 3. In Section 4, we present the Linked NOAA dataset generation with its details, data samples, and query demonstrations. Finally, we also mention several shortcomings of the dataset and related issues in our project in Section 5, and conclude in Section 6.

## 2. Description of Linked NOAA dataset

To minimise the data size, NOAA raw dataset is firstly stored in the ASCII character format. Data field definitions for transmitted elements are provided after this preface, providing definitions of data fields, position number for mandatory data fields, field lengths for variable data fields, minimum/maximum values of transmitted data and values for missing data fields. Secondly, all these files are compressed into tar.gz file and are grouped by year.

The process of converting global weather data from the NOAA dataset involved four main steps: (1) defining the data modelling to represent all the sensors and weather concepts and their relationships in NOAA, (2) decoding and creating the Linked Data triples from the compressed and encoded raw data, (3) enriching the linked data by linking it to other public datasets as Geonames and DBpedia, and (4) storing the dataset in a specific Triple Storage and publishing it by providing a public SPARQL Endpoint.

Table 1

Release details of dataset

| | |
|---|---|
| URL | `http://noaa.graphofthings.org/` |
| SPARQL | `http://noaa.graphofthings.org/sparql/` |
| Licensing | |
| Update | Live update in 1-3 hours |

In the context of the Five Stars of Linked Data Vocabulary Use [2], the Linked NOAA sensor dataset mainly reach 4 stars as follows:

– *There is dereferencable human-readable information about the used vocabulary.* Every information, descriptions, and used vocabularies about the dataset is published at `http://noaa.graphofthings.org/`.
– *The information is available as machine-readable explicit axiomatization of the vocabulary.* The dataset is stored as RDF triples on the Triple Storage.
– *The vocabulary is linked to other vocabularies.* In fact, the dataset is modelled based on the GoT ontology which is extended from the SSN ontology (see also Section 2.1)
– *Metadata about the vocabulary is available.* The information about the license model, contact person, last modification date, and the used ontology language can see from `http://noaa.graphofthings.org/`.

### 2.1. Data modelling

To semantically describe our Linked NOAA dataset, we use some standard ontologies which cover all the concepts and relationships in the sensor data domain. This section will describe in details all the selected ontologies that we have analysed, reused and extended for the describing our Linked NOAA dataset. These ontologies required are as follows:

– Ontologies to describe sensors (e.g., the SSN Ontology, the AWS ontology for Meteorological Sensors).
– Ontologies to describe weather phenomena and their measurable properties (e.g., the Climate and Forecast features Ontology).
– Ontologies to describe units of measurement (e.g., the Quantity Kinds and Units Ontology).
– Ontologies to describe geographical places and their locations (GeoNames and WGS84).

**SSN Ontology.** Semantic Sensor Network ontology[4] is developed by the W3C Semantic Sensor Networks Incubator Group which aims to describe sensors, observations, and related concepts [1]. SSN ontology becomes the most common ontology used in the

[4]`http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/`

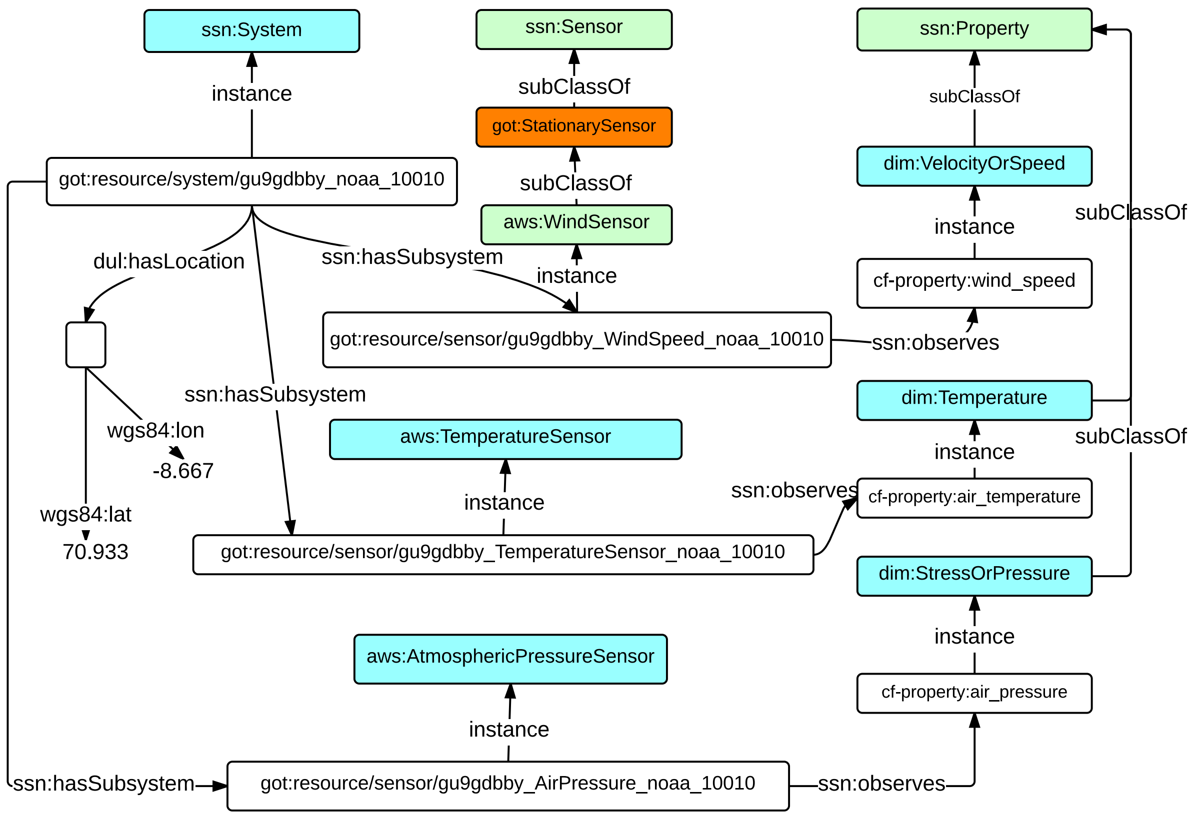


Fig. 1. Sample Of The Ontology For A Sensor Station And Its Observations

sensor network domain. This ontology defines different modules which contains several classes and its relationships to model a system, its sensors, the deployment phases and deployment sites (or platforms).

To model the NOAA weather data, we reuse four main modules of SSN: Skeleton, System, Data and Device. In particular, we focus on these classes:

- **ssn:Observation** to describe an observed feature, an observed property, a sensor and method of sensing used and a value observed for the property.
- **ssn:FeatureOfInterest** to point to the observed feature of interest. A feature of interest can be any observed real-world phenomenon.
- **ssn:Sensor** to describe the sensors.
- **ssn:System** to describe the infrastructure of system with information about its components, subsystems, which are other systems.

For the relationships, we use several main properties correspond to these classes above:

- **ssn:observes** to describe the relation between a Sensor and a Property that the sensor can observe.
- **ssn:observedBy** to describe a sensor which produced the observation (an instance of the class ssn:Sensor).
- **ssn:observedProperty** to describe the specific quality of the feature of interest which was observed (e.g., temperature, acceleration, or speed).
- **ssn:featureOfInterest** to describe the relation between an observation and the entity whose quality was observed.
- **ssn:observationResult** to describe the result of the observation as an instance of the class *ssn:SensorOutput*.
- **ssn:observationResultTime** to describe the time when the observation result became available.

**The AWS Ontology for Meteorological Sensors.** AWS ontology[5] is an ontology for meteorological sensors monitoring Australian weather and agriculture.

[5] `http://www.w3.org/2005/Incubator/ssn/ssnx/meteo/aws`

It mainly describes different models of sensors that related to measuring weather phenomena. This ontology is extended from SSN ontology and its scope cover most of our cases and hence is perfectly match to model our dataset.

Fig. 1 presents the example which combine both ontologies mentioned above. In this example, the system instance, the highest level for a weather sensor station, is an entity of the *ssn:System* class while its subsystem will be an entity of the *aws:WinSensor*, *got:StationarySensor*, and *ssn:Sensor* classes. This inheritance leads to the subsystem instance has properties of *ssn:Sensor*'s properties about sensor features, *aws:WinSensor*'s properties about weather features, and *got:StationarySensor*'s properties about extended features for NOAA dataset and the GoT system as well.

### 2.2. Enrichment

To enrich the NOAA dataset, we use the GeoNames[6] and DBpedia[7] to link weather sensor stations to geographical information. Weather sensor stations in the dataset are linked to their associate GeoNames entities and features via using the GeoNames API. For example, for the weather sensor station which has a location at latitude 70.933 and longitude at -8.667, the enriching step annotates associate features, such as *geonames:parentCountry*, *geonames:parentADM1*, *geonames:parentADM2* (as result in the below sample in Example 1). After the enriching step, the linked NOAA dataset can be queried from city names to sensors or by spatial queries.

```
<got:PhysicalPlace/gu9gdbby>
  a  <dul:PhysicalPlace> ;
   <geonames:nearbyFeatures>
       <http://sws.geonames.org/6954612/nearby.rdf> ;
   <geonames:parentADM1>
       "Jan Mayen" ;
   <geonames:parentADM2>
        <http://sws.geonames.org/7535695/> ;
   <geonames:parentCountry>
       "Svalbard and Jan Mayen" ;
   <geonames:parentCountry>
       <http://sws.geonames.org/607072/> ;
   <http://www.w3.org/2003/01/geo/wgs84_pos#lat>
       "70.933"^^xsd:double ;
   <http://www.w3.org/2003/01/geo/wgs84_pos#long>
       "-8.667"^^xsd:double .
```

Example 1. Enriching Data for Weather Sensor Station location

[6] http://www.geonames.org
[7] http://wiki.dbpedia.org/Datasets

### 2.3. URI scheme definition

Taking into account the analysis that has been done in the previous section about the sufficient ontologies used to model our dataset, there are only few classes (e.g., the class *got:StationarySensor*), that we have had to create. The rest of the ontologies described above is reused from SSN and AWS ontologies. In this section, we will describe the decisions taken in order to create URIs for resources (Section 4.1) and provide a summary of the main types of URL.

URIs have been designed and followed by several general principles, such as simplicity, stability and manageability. We have followed common guidelines for their effective use, following in many suggestions already applied in other existing ontologies on the sensor network domain. In our case, the base URI is `http://graphofthings.org/`, prefixed as ***got***. Hence, all instances follow the URI scheme `http://graphofthings.org/resource/`. For example, the URI to identify the wind sensor of NOAA is http://graphofthings.org/resource/sensor/gu9gdbby_WindSensor_noaa_10010. Table 2 and 3 provide a summary and example of the main types of URIs .

To support geographical querying features of our system **EAGLE** (Elastic Graph Search Engine for Linked Streams), which is described in Section 3, we embedded a Geohash code as part of the URI template that is used to annotate *ssn:System* and *ssn:Sensor*. Geohash is a hierarchical spatial data structure which subdivides space into buckets of grid shape. It offers properties like arbitrary precision and the possibility of gradually removing characters from the end of the code to reduce its size (and gradually lose precision). As a consequence of the gradual precision degradation, nearby places will often (but not always) present similar prefixes. On the other side, the longer a shared prefix is, the closer the two places are. Therefore, this feature enables the system to perform aggregate queries by regions with sufficient response time by calculating all the sub regions that share a same prefix. On top of that, this is also the distinguished feature of our system in comparison with other existing linked data storage systems.

### 2.4. Examples of GoT ontology-based Weather Station Data

In the following subsections, we will be providing examples of how we generate some of our RDF data, according to the selected ontologies.

Table 2
URI generation templates for resources

| Object | Resource class | URL pattern |
|---|---|---|
| Weather Station | ssn:System | <got prefix>_<station type Acronym>_<number ID> |
| Air Temperature | ssn:Sensor | <got prefix>_<station type Acronym>_<number ID> |
| Observation | ssn:Observation | <got prefix>_<property> _<number ID>_<timestamp> |

Table 3
Example of URIs in dataset

| Instance | URI |
|---|---|
| URI station 10010 | got:resource/system/gu9gdbby_noaa_10010 |
| URI temperature sensor | got:resource/sensor/gu9gdbby_TemperatureSensor_noaa_10010 |
| URI observation | got:resource/observation/gu9gdbby_AirTemperature_10010_1964_1_3_23_0 |

**Describing the NOAA Weather Station.** In the first example, we provide a general overview of how we describe the main context of where our weather station is located. Adding to the SSN Ontology (our weather station is an instance of the *ssn:System* class), we use two other vocabularies for this purpose: the W3C WGS84 vocabulary to define the geo-spatial point where the weather station is located, the GeoNames vocabulary in order to relate the weather station to its geometry. Fig. 2 provides a graphical summary of our model and instances.

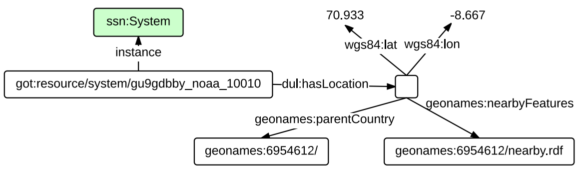


Fig. 2. Description of NOAA 10010 weather station

**Describing the Sensors Deployed in the Weather Station.** Besides describing the Weather Station, by extending the SSN ontology, we have also focused on modelling each of the sensors that are assembled in NOAA weather station. Fig. 3 provides a graphical overview of Temperature sensor which is deployed in NOAA 10010 Weather Station.

Based on the URI scheme definition we mentioned above, this Temperature sensor is identified by an URI that finishes with the term *gu9gdbby_Temperature Sensor_noaa_10010*. An important aspect to note that in order to make the ontology more general and to distinguish with the moving sensor, we add a new class *got:StationarySensor*, which is a subclass of *ssn:Sensor*, and also reuse the *aws:TemperatureSensor* to specify clearly the category of sensor that the TemperatureSensor represents. This TemperatureSensor is a subsystem of the weather station, expressed with the property *ssn:hasSubsystem* between the weather station and the sensor. In the end, we specify that the sensor observes (*ssn:observes*) the property *cf-property:air_temperature* as defined in the Climate and Forecast features Ontology.

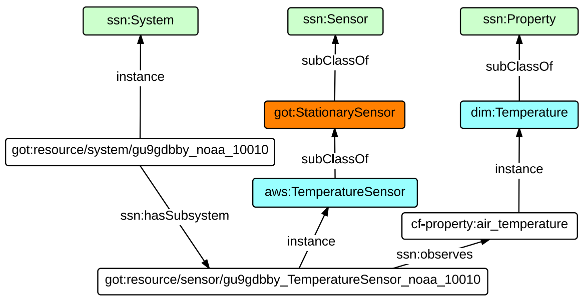


Fig. 3. Description of the Temperature sensor installed in Weather station 10010

**Describing the Observations.** An observation is a situation that describes an observed feature and a value observed for the property. Their description lies at the core of the SSN Skeleton module. The main feature of the Observation is to describes a single value attributed to a single property by a particular sensor. Fig. 4 represents an observation done by the Temperature sensor of NOAA 10010 weather station on the air temperature at a given point in time. The example describes the use of the properties *ssn:observedProperty*, *ssn:featureOfInterest*, *ssn:observedBy*, and *ssn:obser vationResult* related to specific observation with its corresponding observed property.

One of the most important aspect of an observation is to represent the observation values and their corresponding units. As a result of the lack of standard ontology for describing observed values and measurement units, different domains lead to different ways to model observation values. For the sensor network domain, adding to the SSN standard property *ssn:hasValue* is used to relate the result with the corresponding *ssn:ObservationValue*, we also use the most common properties *qu:numericalValue* and *qu:unit* to express the actual value and the unit of measurement that is used for this property.

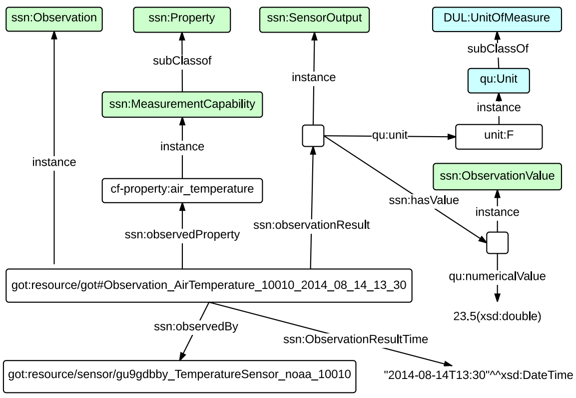


Fig. 4. Description of an observation of air temperature

## 3. Data transformation process and maintenance

The process of generating the Linked NOAA sensor dataset is comprised of three main parts (as shown in Fig. 5). The transformation workflow begins with the NOAA FTP service that provides compressed data in the tar.gz file format. The extraction of these compress files stores the ASCII character textual output in physical storage. Each of output presents sensor observations encoded as tab separated numerical values representing the intensities of phenomena measured by sensor stations. To make the sensor descriptions and observations available in Linked Data format, these observation data need to be converted to RDF. The generated RDF files are then stored in the EGALE system and are open accessed through the public SPARQL endpoint.

**Collecting NOAA data** - This step involves of using a NOAA FTP service to collect hourly update of weather data and decoding the result. NOAA provides a public FTP service to access historical weather data which are grouped by year and stored in separated folders. These folders include all the observation data files which have been observed in same year and take a station ID, year number as parts of the file name tem-

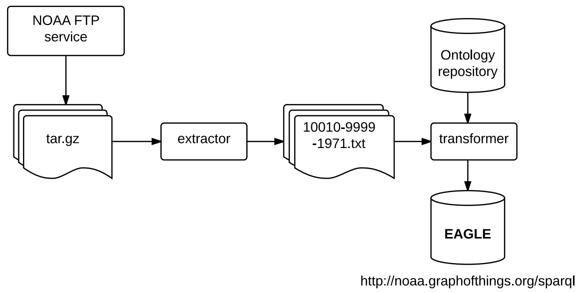


Fig. 5. The Process of Data Transformation

plate. Since we already know the update rate, we build a Java process to automatically access and collect the latest update data from NOAA. The data is then extracted and decoded to textual sensor observation data. Fig. 6 shows a snapshot of sample of NOAA's raw data file.

**Converting to Linked Data** - This is the second step which consists of building a mapping process to convert the raw textual data retrieved from NOAA to RDF. The mapping script is implemented by using custom build XSLT and Jena[8]. While XSLT is a language for transforming XML documents into other XML documents, Jena is an open source Semantic Web framework for Java to provide an API to extract data from and write to RDF graphs. This step raises challenges for converting a large amount of raw data into RDF, which could be a time consuming problem if the process is run in a single thread. From this point forward, we implement a transformer which can be executed in distributed systems by using a Map/Reduce framework[9]. The output of the transformer is RDF data (177 billion triples) based on the GoT ontologies which are described in Section 2.1.

**Storing and publishing dataset through SPARQL endpoint** - The final step is storing the RDF into a triple storage and providing a SPARQL endpoint for the dataset. However, providing a public access service for exploring and querying such data at real-time is extremely challenging due to its Big Data natures: big volume, fast real-time update and messy data sources. To address this challenge we provides a scalable and elastic engine (**EAGLE**) to deal with billion records of historical and static datasets in conjunction with millions of triples being fetched and enriched per hour at real-time. The dataset is accessible via `http://noaa.graphofthings.org/`

[8] `https://jena.apache.org`
[9] `http://hadoop.apache.org/`

```
| USAF  WBAN YR--MODAHRMN DIR SPD GUS CLG SKC L M H  VSB MW MW MW MW AW AW AW AW W TEMP DEWP    SLP
109340 ***** 197101010300 020  7 *** 722 SCT 5 0 0  5.0 10 ** ** ** ** ** ** ** 0  12   9 1011.7 ***** ****** *** *** ***** ***** ***** ***** **
109340 ***** 197101010600 030  7 *** 722 SCT 6 0 0  5.0 10 ** ** ** ** ** ** ** 0  10   5 1012.6 ***** ****** ***  10 ***** ***** *****  0.02 **
109340 ***** 197101010900 030  6 *** 722 SCT 5 7 1  3.8 10 ** ** ** ** ** ** ** 7  10   7 1014.7 ***** ****** *** *** ***** ***** ***** ***** **
109340 ***** 197101011200 060  5 *** 722 SCT 2 4 0  6.2 02 ** ** ** ** ** ** ** 7  23  14 1014.0 ***** ****** *** *** ***** ***** *****  0.02 **
109340 ***** 197101011500 050  5 *** 722 SCT 5 0 0  6.9 01 ** ** ** ** ** ** ** 0  21  14 1015.8 ***** ****** *** *** ***** ***** ***** ***** **
109340 ***** 197101011800 020  7 *** 722 SCT 5 0 0  5.6 03 ** ** ** ** ** ** ** 0  16  10 1019.1 ***** ******  25 *** ***** ***** *****  0.02 **
109340 ***** 197101012100 020  7 ***  23 BKN 5 * *  3.1 10 ** ** ** ** ** ** ** 1  14   9 1020.9 ***** ****** *** *** ***** ***** ***** ***** **
109340 ***** 197101020000 020  8 *** 722 CLR 0 0 0  2.5 10 ** ** ** ** ** ** ** 1   9   5 1021.7 ***** ****** *** *** ***** ***** ***** ***** **
109340 ***** 197101020300 020  7 *** 722 CLR 0 0 0  2.5 10 ** ** ** ** ** ** ** 0   5   3 1023.5 ***** ****** *** *** ***** ***** ***** ***** **
```

Fig. 6. Sample Of A NOAA's Raw Data File

`sparql` which will be described in details in the following section.

## 4. Application on NOAA Linked Sensor Data

### *4.1. Access to Linked NOAA dataset by public SPARQL Endpoint*

As our global weather dataset is a Linked Dataset, we make it accessible via SPARQL endpoint at `http://noaa.graphofthings.org/sparql/`. However, this SPARQL endpoint supports more powerful SPARQL query language than SPARQL 1.1. For spatial computation, it supports spatial extension for SPARQL query via Jena Spatial built-in function which are mapped to spatial computation functions provided by Blur. For query graph pattern associated with time series data, we also support temporal extension for SPARQL which is backed by our modified version of Open TSDB[10]. On top of that, full text search is supported by fuzzy matching syntaxes of Lucence[11] which is processed by Blur[12]. Following are several examples of a SPARQL queries over Linked NOAA dataset using spatial, temporal and free-text search patterns.

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  text: <http://jena.apache.org/text#>
PREFIX  spatial: <http://jena.apache.org/spatial#>
PREFIX  temporal: <http://jena.apache.org/temporal#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
SELECT *
{?loc spatial:withinCircle (67.033 -178.917 10.0
                                         'miles' 100).
        ?loc geo:lat ?lat.
        ?loc geo:long ?long.
        ?sensor dul:hasLocation ?loc.
}
```

Example 2. Spatial query for sensor discovery

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  text: <http://jena.apache.org/text#>
PREFIX  spatial: <http://jena.apache.org/spatial#>
PREFIX  temporal: <http://jena.apache.org/temporal#>
SELECT *
{
   ?value temporal:avg ('1971/01/01-03:00'
           '1971/01/01-09:00' 'groupin' '1h-avg'
           'u' 'temperature').
}
```

Example 3. Aggregate query for sensor historical data

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  text: <http://jena.apache.org/text#>
PREFIX  spatial: <http://jena.apache.org/spatial#>
PREFIX  temporal: <http://jena.apache.org/temporal#>
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
SELECT *
{?loc spatial:withinCircle (67.033 -178.917 10.0
                                      'miles' 100).
        ?sensor dul:hasLocation ?loc.
        ?sensor ssn:observes ?property.
        ?sensor temporal:sensors('1971/01/01-03:00'
                                      '1971/01/01-09:00').
        ?sensor text:query (rdfs:label 'Birmingham').
}
```

Example 4. Query using spatial temporal and free-text search patterns

### *4.2. Sensor Discovery and Data Mashup application*

We have built a GraphOfThings[13] application that provides an easy-to-use map-based GUI in order to discover all kind of sensors at the specific location [3]. The application uses the NOAA Linked Weather Data as a sub dataset combines with our other datasets. The application provides to users the geographical functionalities from which the users can choose the area that they are interested in by providing a location name in the search box. The application then will generate and execute a SPARQL query over the dataset and displays all the sensors near the given lo-

[10] `http://opentsdb.net`
[11] `http://lucene.apache.org/core/`
[12] `https://incubator.apache.org/blur/`
[13] `http://graphofthings.org`

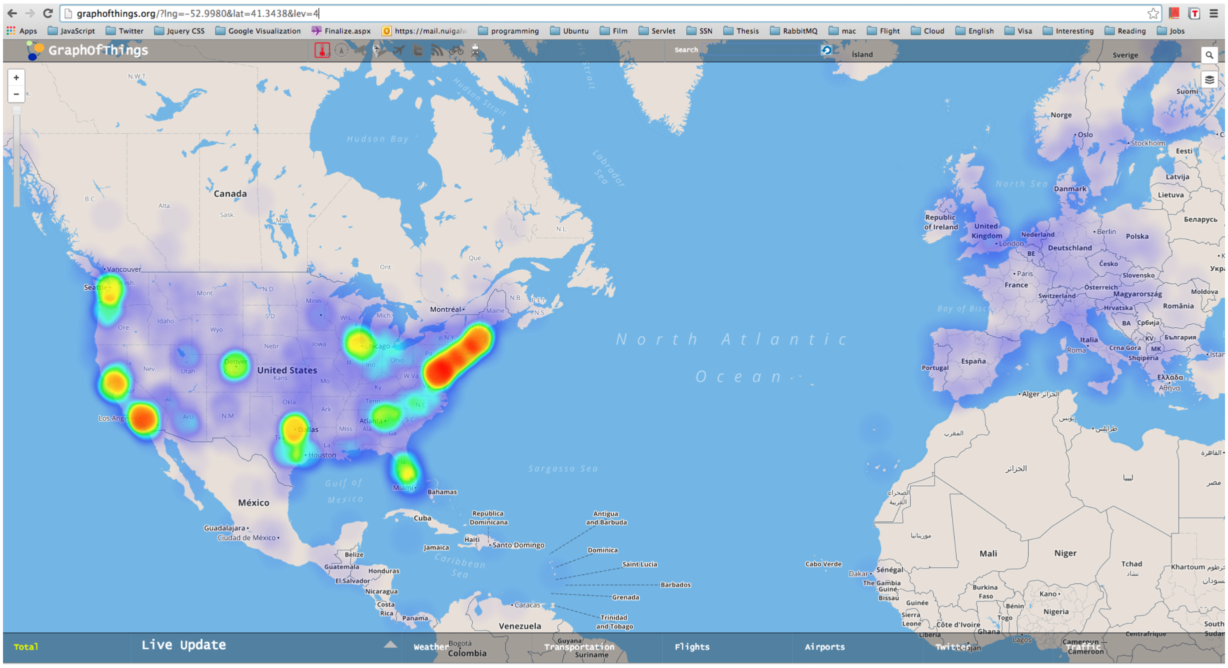


Fig. 7. Screen shot of GraphOfThings application

cation. The sensors displayed on the map also provide additional information such as phenomena measured and geographic coordinates. In addition, the sensor descriptions also present the live data which are being streamed from **EAGLE** system. The visualisation functionality provides an analytical view of sensor data over times. A screenshot of the application can be found in Figure 7.

## 5. Shortcomings

This linked dataset has a tendency to observations rather than information about weather systems. In fact, NOAA's dataset has a metadata file to describe sensor stations but there is a lack of description of configuration of weather sensor stations from NOAA's dataset. Other point is the update rate of the dataset which is quite slow at from 1 to 3 hours.

Moreover, there is a determined limitation in the system performance when implementing the linked dataset on our system. Currently, our triple storage is still under development and the hardware of the system is limited while the amount of sensor data output is tremendous. In addition, the engine needs to have further development to deal with queries that are too complicated.

## 6. Conclusion

The paper presents the process of creating Linked NOAA dataset from weather dataset which is made publicly available by NOAA via FTP protocol. Employing linked data publishing mechanism, NOAA dataset is made available to query via a SPARQL endpoint as a part of Link Data Cloud.

In summary, the contribution of the paper is listed as follows:

– Modelling the Linked NOAA Dataset based on GoT ontology for weather sensor stations. This ontology is extended from the SSN Ontology and the AWS ontology for Meteorological Sensors.
– Providing an available linked dataset, namely Linked NOAA Dataset, for weather sensor stations over the world from 1901 until present. This dataset with over 177 billion triples is stored in the Triple Storage server and published at `http://noaa.graphofthings.org/`. It also can be accessed via SPARQL endpoint at `http://noaa.graphofthings.org/sparql/`.
– Our project also builds a GraphOfThings application based on map GUI approach in order to discover all kind of sensors, including weather sensors from the Linked NOAA Dataset.

This application is public at `http://noaa.graphofthings.org/sparql/`.

– Describing a completed process to build the global linked sensor dataset from the raw NOAA data files, including the modelling stage, URI scheme definition, Data transformation process, and its applications.

## References

[1] Michael Compton, Payam Barnaghi, Luis Bermudez, RauL GarciA-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25–32, 2012.

[2] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman II. Five stars of linked data vocabulary use. *Semantic Web*, 5(3):173–176, 2014.

[3] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Quoc Hung Ngo, Tuan Tran Nhat, and Manfred Hauswirth. Enabling live exploration on the graph of things. *Proceedings of the Semantic Web Challenge 2014, Organised in conjunction with the International Semantic Web Conference*, 2014.