A Real-time Integration of Semantics into Heterogeneous Sensor Stream Data with Context in the Internet of Things

Besmir Sejdiu[1][a], Florije Ismaili[1][b] and Lule Ahmedi[2][c]
[1]*Contemporary Sciences and Technologies, South East European University, Tetovo, Macedonia*
[2]*Faculty of Electrical and Computer Engineering, University of Prishtina, Prishtinë, Kosova*

Keywords: Sensor Stream Data, Semantic Annotations, Semantic Sensor Web (SSW), Internet of Things (IoT).

Abstract: Recently, billions Internet of Things (IoT) devices, including sensors are producing sensed data continuously in the stream data, and transmit these data to a centralized server. Due to the dramatically increase of streaming data, their management and exploitation has become increasingly important and difficult to process and integrate the semantic to sensor stream data in real-time. This research focuses on real-time integration of semantics into heterogeneous sensor stream data with context in the IoT. In this context, an IoT real-time air quality monitoring system and different semantic annotations are developed for sensor stream data in the format of Sensor Observation Service (SOS).

1 INTRODUCTION

The Internet of Things (IoT) represents an active scientific research field due to its importance in development of many domains, including environmental monitoring, healthcare, homes, cities, traffic control, energy systems, industry, etc. Sensors are one of the main components that enable IoT, which send the observation in stream data.

Furthermore, sensor data are enabled to the web through the Sensor Web. Sensor Web by incorporating technologies of the Semantic Web creates the Semantic Sensor Web. In this way, sensor data stream can be annotated with semantics by providing machine-interpretable descriptions on what the data represents, where it originates from, how it can be related to its surroundings, who is providing it, and what are the quality, technical, and non-technical attributes (Barnaghi, 2012). The real-time integration of sensor data as dynamic data with semantics is defined as *real-time semantic annotation*, while sensor data that are stored in repository (data store) as static data, and then integrated with semantics is defined as *non-real-time semantic annotation* (Sejdiu, 2020).

Organizations like Open Geospatial Consortium (OGC) and World Wide Web Consortium (W3C) have proposed several standards for sensor data. The OGC defines standardization for the Sensor Web named Sensor Web Enablement (SWE). It's a framework and a set of standards that allow exploitation of sensors and sets of sensors connected to a communication network. Is founded on the concept of "Web Sensor" using standard protocols and application interfaces (Pradilla, 2016).

In this paper, we will investigate on how to integrate semantic annotations into the sensor stream data. In particular, we will discuss the annotation techniques for real-time integration of semantics into heterogeneous sensor observation data and sensor metadata with context in the IoT.

The paper is organized as follows: Section II provides a discussion on related work for semantic annotations to the sensor stream data. Section III is an overview of the sensor stream data and semantic annotations concepts. Selection of technologies and standards for semantic annotations are presented in Section IV, while a system architecture is presented in Section V. Section VI represents the implemented system, including received sensor data format, integration of semantic annotations to the sensor data,

[a] https://orcid.org/0000-0002-2786-5384
[b] https://orcid.org/0000-0002-3627-0147
[c] https://orcid.org/0000-0003-0384-6952

Sejdiu, B., Ismaili, F. and Ahmedi, L.
A Real-time Integration of Semantics into Heterogeneous Sensor Stream Data with Context in the Internet of Things.
DOI: 10.5220/0009884403760383
In *Proceedings of the 15th International Conference on Software Technologies (ICSOFT 2020)*, pages 376-383
ISBN: 978-989-758-443-5

and system outputs. Finally, Section VII concludes the paper and identifies some of the future perspectives of the semantic integrations into the sensor stream data.

2 RELATED WORK

Authors in (Aggarwal, 2013) brought together semantic web and data mining in the context of IoT with a focus on sensors as interconnected devices, concluding that practical data mining applications can be built by usage of real world sensors ontologies, query mechanisms and linked sensor data available. SSW is described as a synthesis of sensor data and semantic metadata in (Sheth, 2008). It represents an approach by OGC and Semantic Web Activity of the W3C to provide meaning for sensor data. Construction of a Semantic Sensor Observation Service (SemSOS) based on the SWE standards is discussed in (Henson, 2009), by adding semantic annotations to sensor data and by using the ontology models to reason over sensor observations.

An extension of the SWE framework in order to support standardized access to sensor data is described in (Lee, 2015). Furthermore, they list as future work the extension of SOS server with semantics, since the lack of semantically rich mechanism is seen as a significant issue, which makes it hard to explore related concepts, subgroups of sensor types, or other dependencies between the sensors and data collected.

The objective of this paper is to present a real-time integration of semantics into heterogeneous sensor stream data with context in the IoT.

3 SENSOR STREAM DATA AND SEMANTIC ANNOTATIONS

IoT applications are enabled using heterogeneous sensors, which send observational data referred to as sensor stream data to a remote server. Raw sensor stream data is useless unless properly annotated. Therefore, the researchers proposed Semantic Sensor Web (SSW), which is a combination of Sensor Web and technologies of Semantic Web. Based on study (Sejdiu, 2020), the explored publications show that major number of research are accepting the proposed industry standards, such as SWE, and techniques that can be used for annotating sensor data, such as Resource Description Framework in attributes (RDFa), XML Linking Language (Xlink), and Semantic Annotations for WSDL and XML Schema (SAWSDL), by different organizations like OGC and W3C. However, how to advance techniques for integration of the semantic annotations in real-time is still an open issue that should be addressed.

4 SELECTED TECHNOLOGIES AND STANDARDS

Currently, billions of interconnected IoT devices produce sensed data continuously in the stream data, and transmit these data to a centralized server. Due to the dramatically increase of streaming data, their management and exploitation has become increasingly important and difficult to process and integrate the semantic to sensor data stream in real time. Therefore, the selection of technologies and standards for technique development of real-time integration of semantics into heterogeneous sensor observation data and sensor metadata with context in the IoT is highly important. The proposed real-time semantic annotation system utilizes Spark Streaming[1], Apache Kafka[2], Apache Cassandra database[3], and standards like OGC Sensor Web Enablement standards, which will be discussed below.

4.1 Spark Streaming

Several stream data processing systems including Spark Streaming, Storm, Google Data Flow, and Flink have emerged to support real-time analytics for the streaming data sets (Karimov, 2018). Majority studies conclude that Spark Streaming works best with high throughput when the incoming volume is huge (Gorasiya, 2019). Therefore, we have chosen Sparking Streaming to develop our system for real-time integration of semantic annotations to sensor stream data.

Spark Streaming is an extension of the Apache Spark that enables to build scalable fault-tolerant IoT applications for real-time processing sensor stream data. It can receive data from different input sources such as Apache Kafka, TCP sockets, Flume, Kinesis, Hadoop Distributed File System (HDFS), or Twitter, and can be processed using complex algorithms expressed with high-level functions like *map*, *join*,

[1] http://spark.apache.org
[2] https://kafka.apache.org
[3] http://cassandra.apache.org

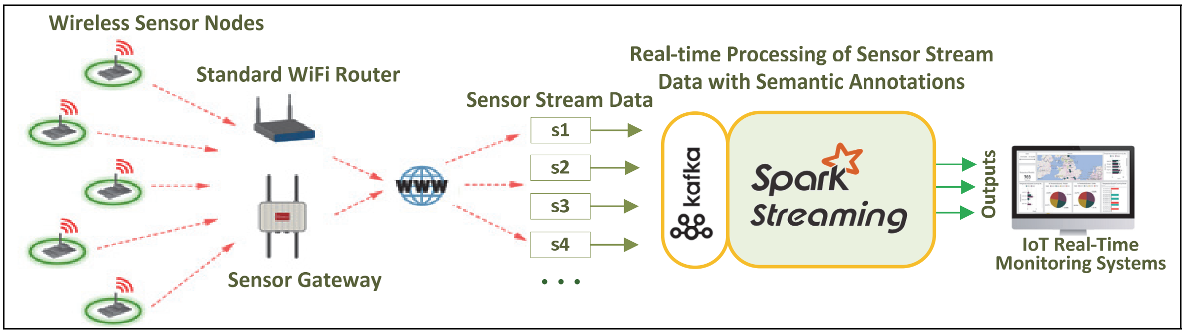

Figure 1: An overview of the system architecture.

reduce and *window*. Finally, processed streaming data can be published in IoT real-time applications or can be pushed out to databases or file systems.

4.2 Apache Kafka

Apache Kafka is a distributing streaming platform with capabilities to publish and subscribe to streams of records, similar to a message queue or enterprise messaging system, store streams of records in a fault-tolerant durable way, and process streams of records as they occur. Kafka is generally used for building real-time streaming data pipelines that reliably get data between systems or applications (Kafka, 2020). In our system Kafka is used as middleware between sensor stream data and Spark Streaming.

4.3 Apache Cassandra Database

The Apache Cassandra database is a free and open source, distributed store for structure data that scale-out on cheap, commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. It is designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. The Spark Streaming interacts well with Cassandra database. Therefore, in our system, the sensor stream data with their semantic annotations processed by Spark Streaming are stored in Cassandra database.

4.4 OGC Standards

The OGC defines standardization for the Sensor Web named Sensor Web Enablement (SWE), which is divided into two parts (OGC, 2020):

- *SWE Information Model*: Is comprised of conceptual language encodings that permits sensor observations visibility on the Internet. The SWE information model includes the following specifications: Sensor Model Language (SensorML), Observation and Measurement (O&M), and Transducer Model Language (TransducerML).
- *SWE Service Model*: Is a set of Web Service specifications that allow a client to search and find the required information. The SWE Service model includes the following specifications: Sensor Observation Service (SOS), Sensor Alert Service (SAS), Sensor Planning Service (SPS), and Web Notification Services (WNS).

To encode semantic annotations and data gathered by sensors, in this paper is used SOS O&M, which will be discussed in section 6.2.

5 SYSTEM ARCHITECTURE

In the Figure 1, an overview of the system architecture for real-time integration of semantics into heterogeneous sensor stream data with context in the Internet of Things is presented. As mentioned above, the proposed real-time semantic annotation system utilizes Apache Kafka, Spark Streaming, Apache Cassandra database, and SOS O&M standards.

The heterogeneous sensor stream data from the IoT-based sensor device is wirelessly transmitted to serve as the “producer” for the Kafka server. The “producer” client publishes streams of data to Kafka “topics” distributed across one or more cluster nodes/servers called “brokers”. The published streams of data from Kafka are then processed by Apache Spark Streaming in parallel and real-time.

Kafka server is utilized to receive various formats of sensor data streams (e.g. text, binary, JSON, XML etc.), and to transform them in a particular format that will be processed by Spark Streaming.

The Spark Streaming enables a real-time integration of semantics into heterogeneous sensor stream data with context in the IoT, by using sensor metadata, archival data streams, mining data streams, association rules for adding semantic annotations with concept definitions from ontologies or other semantic sources, which allows the understanding of senor data and metadata elements. The semantic annotations will be implemented into SOS O&M by using stakes, such as XLink (without including XPath) and Embedded (only a single value-scalar of semantic annotation) to add annotations in XML files. These annotations can point to extra sources of information (e.g. a file), or Uniform Resource Name (URN).

The enriched sensor stream data with the semantic annotations results will be stored in the Cassandra database, and will be displayed in IoT real-time monitoring system. It is worth mentioning that Spark Streaming will process sensor data stream in format of OGC standards like SWE, respectively version 2.0 of the SOS standard to encode semantic annotations and data gathered by sensors (Bröring, 2012).

The detailed description is presented in section 6.2 where an example of integration of semantic annotations into the sensor stream data with context in the IoT is given.

6 IMPLEMENTED SYSTEM

An IoT real-time air quality monitoring system is developed to visualize sensor stream data and their semantic annotations, based on web platform. Sensor data of Hydrometeorological Institute of Kosovo (HMIK[4]) are used, through World Air Quality Index API (AQI API). The AQI API can be used for advanced programmatic integration, such as: access to more than 11000 station-level and 1000 city-level data, station name and coordinates, search station by name, geo-location query based on latitude/longitude, individual Air Quality Index (AQI) for all pollutants, current weather conditions, etc (Aqicn, 2020).

6.1 Received Sensor Stream Data Format

The system receives raw sensor stream data from AQI API in JSON format, as presented in Figure 2, which supports measuring in real-time of the following parameters: Carbon Monoxide (co), Humidity (h),


```
{
"status": "ok",
"data": {
  "aqi": 58,
  "idx": 12402,
  "attributions": [
    {
     "url": "http://worldweather.wmo.int",
     "name":"World Meteorological Organization
- surface synoptic observations (WMO-SYNOP)"
    },
    {
      "url": "http://ihmk-rks.net/",
      "name": "Instituti Hidrometeorologjik i
Kosovës",
      "logo": "Kosovo-IHMK.png"
    },
    {
     "url": "https://waqi.info/",
     "name": "World Air Quality Index Project"
    }
  ],
  "city": {
    "geo": [ 42.648872, 21.137121 ],
    "name": "Prishtine - IHMK, Kosovo",
    "url":
"https://aqicn.org/city/kosovo/prishtine-ihmk"
  },
  "dominentpol": "pm25",
  "iaqi": {
    "co": { "v": 33.3 }, "h": { "v": 76 },
    "no2": { "v": 6.2 }, "o3": { "v": 23.3 },
    "p": { "v": 1015.7 }, "pm10": { "v": 17 },
    "pm25": { "v": 58 }, "so2": { "v": 6.3 },
    "t": { "v": 1.6 },"w": { "v": 14 },
    "wg": { "v": 23 }
  },
  "time": {
    "s": "2020-03-25 20:00:00",
    "tz": "+01:00", "v": 1585166400
  },
  "debug": { "sync": "2020-03-
26T04:17:09+09:00" }
}}
```


Figure 2: Sensor stream data - JSON format.

Nitrogen Dioxyde (no2), Ozone (o3), Pressure (p), PM_{10} (pm10), PM_{25} (pm25), Sulphur Dioxide (so2), Temperature (t), Wind (w), and Water Gauge (wg). As shown in Figure 2, JSON data contains also attributes such as: *data* (station data: *idx* - unique id for the city monitoring station, *aqi* - real time air quality information, *time* - measurement time information, *s* - local measurement time, and *tz* - station time zone), *city* (information about the monitoring station: *name* - name of the monitoring station, *geo* - latitude/longitude of the monitoring station, and *url* - url for the attribution link), *attributions* (EPA Attribution for the station), and *iaqi* (measurement time information: *pm25* - individual AQI for the PM2.5, *v* - individual AQL for the PM2.5).

[4] http://ihmk-rks.net/

Data received by sensors every 6 minutes, through AQI API, are represented in corresponding numerical formats, e.g. in -3.8 (°C) for temperature parameter.

6.2 Integration of Semantic Annotations to the Sensor Stream Data

In our system, different semantic annotations for sensor stream data are developed, such as:

- *#AIQ_Index*,
- *#Air_Pollution_Level*, and
- *#Health_Implications*

#AIQ_Index annotation – is an index for reporting daily air quality, and tells how clean or polluted air is. United States Environmental Protection Agency (EPA[5]) calculates the AQI for five major air pollutants regulated by Clean Air Act: ground-level ozone, particle pollution (also known as particulate matter), carbon monoxide, sulfur dioxide, and nitrogen dioxide. The AQI range values is from 0 to 500. According to EPA, the higher the AQI value, the greater the level of air pollution and the greater the health center (take the maximum of all individual AQI), as presented equation 1:

$$AQI = \max(AQI_{PM2.5}, AQI_{PM10}, AQI_{O3}, ...) \qquad (1)$$

#Air_Pollution_Level annotation – based on the AQI value, its divided into six '*Air Quality Index Levels of Health Concern'* categories:

- *Good* (AQI is 0 to 50)
- *Moderate* (AQI is 51 to 100)
- *Unhealthy for Sensitive Groups* (101 to 150)
- *Unhealthy* (AQI is 151 to 200)
- *Very Unhealthy* (AQI is 201 to 300)
- *Hazardous* (AQI is 301 to 500)

#Health_Implications annotation – each of six categories described above, corresponds to a different level of health concert. *#Health Implications* annotation tells what they mean, for example *"Unhealthy for Sensitive Groups"* category means: 'Although general public is not likely to be affected at this AQI range, people with lung disease, older adults and children are at a greater risk from exposure to ozone, whereas persons with heart and lung disease, older adults and children are at greater risk from the presence of particles in the air.', or for *"Moderate"* category: 'Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.'

The above described annotations are developed into ontology named '*ont-core'*.

After describing different types of the semantic annotations for sensor stream data, in the following is presented the process of semantic annotations.

The sensor stream data may arrive in different formats to Kafka server (JSON format - in our case), which will transform them in a specific format that will be processed by Spark Streaming. After that, through the Spark Streaming, based on measuring values, the sensor data stream will semantically be annotated and converted in SOS O&M format. A fragment of an example of integrated semantic annotations to the SOS O&M format by using stakes like XLink and Embedded, is presented in Figure 3.

SOS O&M observation document comprise zero or multiple *observationData* entries, and each store an instance of an observation. In the following are presented common observation properties (the prefix gml indicates that this element is defined in OGC 07-033, while the prefix om indicates that the element is defined in OGC 10-025r1) (Jirka, 2014):

- *gml:identifier* (mandatory): identifies or refers to a specific observation.
- *om:phenomenonTime* (mandatory): describes the time instant or time period for which the observation contains sensor data.
- *om:resultTime* (mandatory): provides the time when the result became available (often this is identical to the phenomenonTime).
- *om:procedure* (mandatory): the identifier of the sensor instance that has generated the observation.
- *om:observedProperty* (mandatory): the identifier of the phenomenon that was observed.
- *om:featureOfInterest* (mandatory): an identifier of the geometric feature (e.g. sensor station) to which the observation is associated.
- *om:result* (mandatory): the observed value, the type of the result is restricted to the types such as: *gml:MeasureType, xs:integer, xs:boolean, gml:ReferenceType, xs:string, swe:DataArray*.

We have developed a new type of observation to add, named '*SemObservation*' with '*gml:Sem MeasureType*' result type, as shown and described in Table 1.

[5] https://www.epa.gov

```
<sos:Observation ...="">
<observationData>
  <om:OM_Observation gml:id="o23525">
    <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/>
    <om:phenomenonTime>
      <gml:TimeInstant gml:id="phenomenonTime_1">
        <gml:timePosition>2020-03-25T20:00:00+09:00</gml:timePosition>
      </gml:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime xlink:href="#phenomenonTime_1"/>
    <om:procedure xlink:href="http://myserver/ontologies/ont-core.owl#Sensor562415"/>
    <om:observedProperty xlink:href="http://myserver/ontologies/ont-core.owl#PM25"/>
    <om:featureOfInterest xlink:href="http://myserver/ontologies/ont-core.owl#Prishtine"/>
    <om:result xsi:type="gml:SemMeasureType" uom="pm25">
      <value>58</value>
      <sem-annotations>
        <annotation xlink:href="http://myserver/ontologies/ont-
core.owl#Air_Pollution_Level_Moderate"/>
        <annotation embedded:AIQ_Index ="58"/>
        <annotation xlink:href="http://myserver/ontologies/ont-
core.owl#Health_Implications_Moderate"/>
      </sem-annotations>
    </om:result>
  </om:OM_Observation>
</observationData>
. . .
</sos:Observation>
```

Figure 3: An example of integrated semantic annotations to the sensor stream data.

Table 1: The developed *SemObservation* observation type.

Observation Type	Result Type	Description	Example
SemObservation	*gml:SemMeasureType*	Inside the result element, two children elements will be added: *value* and *sem-annotations*. The *value* element will contain a scalar numerical value, while the *sem-annotations* element will contain one or more annotation empty elements.	*<om:result xsi:type="gml:SemMeasureType" uom="pm25">* *<value>58</value>* *<sem-annotations>* *<annotation xlink:href="http:// myserver/ontologies/ont-core.owl#Air_ Pollution_Level_Moderate"/>* *<annotation embedded:AIQ_Index ="58"/>* *<annotation xlink:href="http://myserver/ontologies/ont-core.owl#Health_Implications_Moderate"/>* *</sem-annotations> </om:result>*

For clearer explanation of semantic integration to sensor observation data, Figure 4 illustrates (a) the concept of the O&M and relationship between the entities involved in observations, (b) data streams generated from wireless sensor networks, (c) the sensor data integrated with sensor metadata, archival data streams and the ontological knowledge, and finally, (d) the semantic annotated data with attributes: sem-annotations data, the observed value, unit, metadata, location, timestamp, result type, and gml:id of observation.

6.3 System Outputs

To display the heterogeneous sensor stream data and their semantic annotations, is developed an real time IoT application in the *ASP.NET Core MVC*, a cross-platform, high-performance, open source framework for building modern, cloud-based, and Internet-connected applications. The '*DataStax C# for Apache Cassandra*' is used to read data from Apache Casandra database. It's a modern, feature-rich and highly tunable C# client library. To display

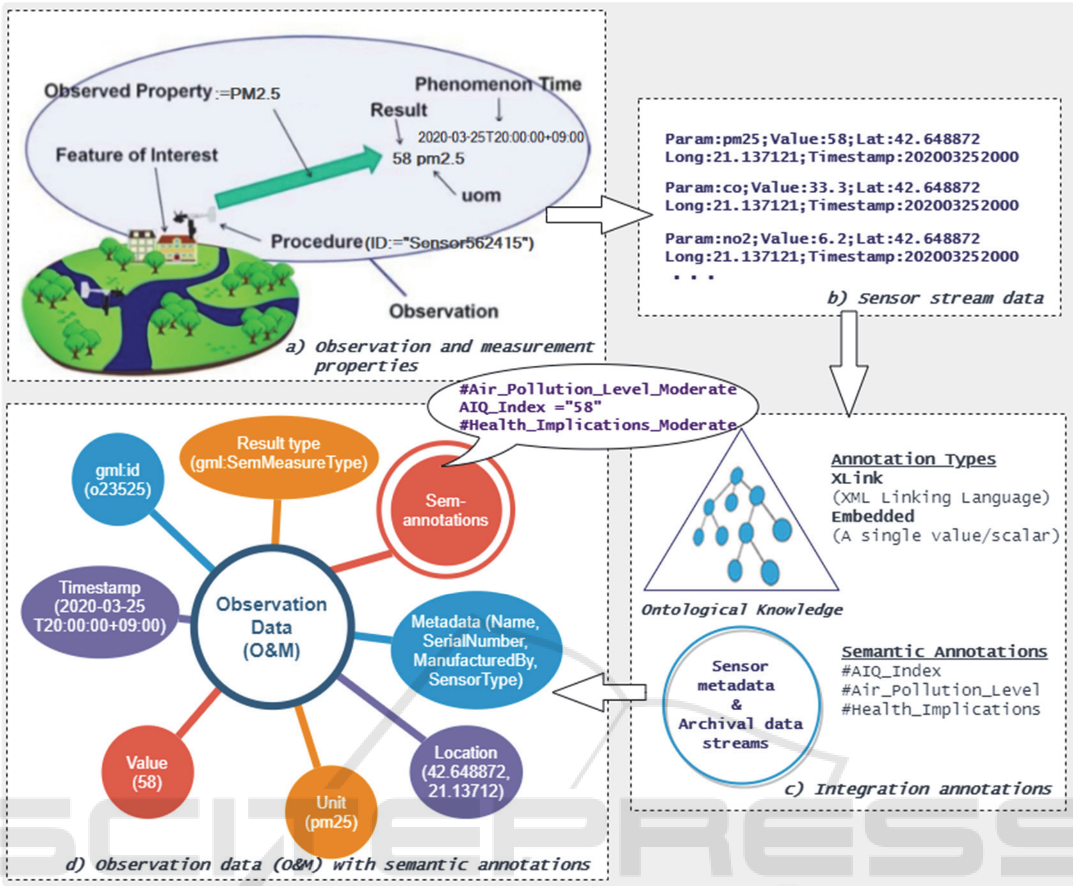

Figure 4: Integrating semantics to sensor observation data.

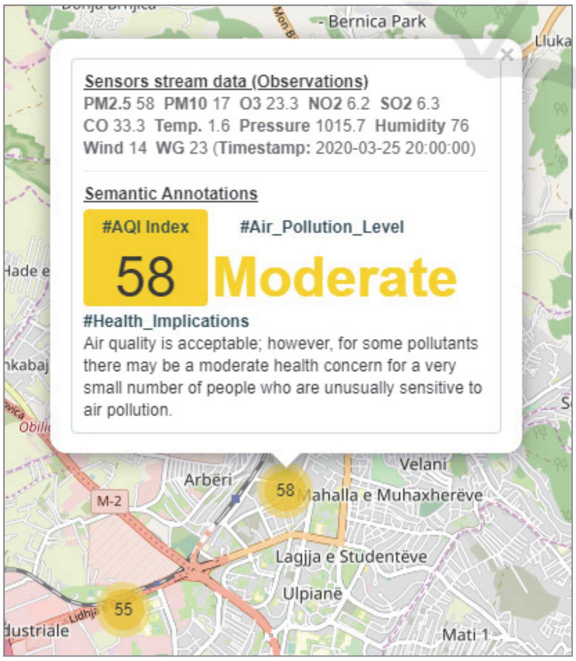

Figure 5: System Outputs.

the data in the map, is used *Leaflet*, an open-source JavaScript library for interactive web maps. *Leaflet* is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms out of the box, taking advantage of HTML5 and CSS3 on modern browsers while being accessible on older ones too.

As shown in Figure 5, the users can observe the quality of air pollution on certain geographical points in map marked as measuring nodes. Each node (marker) has an AQI Index, to indicate air pollution. When clicking over a whatever marker, the latest measurement values obtained for that point will be shown, such as: PM2.5, PM10, O3, NO2, SO2, CO, Temperature, Pressure, Humidity, Wind, Water Gauge, and semantic annotations, such as: #AQI Index, #Air Pollution Level, and #Health Implications.

7 CONCLUSIONS

The IoT represents an active scientific research field due to its importance in different domain applications. Sensors are one of the most important components of the IoT. Raw sensor stream data are useless unless properly annotated. Therefore, by adding semantic annotations with concept definitions from ontologies, it's possible the interpretation and understanding of sensor stream data.

This study presents a system for real-time integration of semantics into heterogeneous sensor stream data with context IoT. First, selected technologies and standards for semantic annotations, such as Spark Streaming, Apache Kafka, Apache Cassandra, and OGC standards are described. Then, the system architecture and implementation of an IoT real-time air quality monitoring system is presented, including:

a) Received sensor data in JSON format of the following measuring parameters: *co*, *h*, *no2*, *o3*, *pm10*, *pm25*, *so2*, *t*, *w*, and *wg*,

b) Integration of semantic annotations such as *#AIQ Index*, *#Air Pollution Level*, and *#Health Implications* to the sensor stream data in SOS O&M format. One of the most important point of this research is that a new type of observation *SemObservation* (with *gml:Sem MeasureType* result type) is developed, and

c) System outputs to display the heterogeneous sensor stream data and their semantic annotations.

Extending the system with more advanced real-time annotation techniques of semantics such as XPath annotations and development of techniques for real-time interpretation of semantic annotations is left for future work.

REFERENCES

Aggarwal, C. C., Ashish, N., & Sheth, A. (2013). The internet of things: A survey from the data-centric perspective. *In Managing and mining sensor data*. Springer US. (383-428).

Aqicn, API – Air Quality Programmatic APIs, [Online, Accessed 20/02/2020]. Available: https://aqicn.org/api.

Barnaghi, P., Wang, W., Henson, C., & Taylor, K. (2012). Semantics for the Internet of Things: Early Progress and Back to the Future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1), 1-21.

Bröring, A., Stasch, C., & Echterhoff, J. (2012). OGC sensor observation service interface standard. *Open Geospatial Consortium Interface Standard*, 12-006.

Gorasiya, D. V., 2019. Comparison of Open-Source Data Stream Processing Engines: Spark Streaming, Flink and Storm. *Technical Report*, DOI: 10.13140/RG.2.2.16747.49440.

Henson, C. A., Pschorr, J. K., Sheth, A. P., & Thirunarayan, K. (2009). SemSOS: Semantic sensor observation service. In *International Symposium on Collaborative Technologies and Systems, 2009. CTS'09*. IEEE. (44-53).

Jirka, S., Stasch, Ch. & Bröring, A. (2014). OGC Best Practice for Sensor Web Enablement, Lightweight SOS Profile for Stationary In-Situ Sensors. *Open Geospatial Consortium.* Version 1.0, ref. no. 11-169r1.

Kafka Apache, Kafka Apache – A distributed streaming platform, [Online, Accessed 15/02/2020]. Available: https://kafka.apache.org.

Karimov, J., Rabl, T., Katsifodimos, A., Samarev, R., Heiskanen, H., & Markl, V., 2018. Benchmarking Distributed Stream Data Processing Systems. In *Proceedings of the IEEE 34th International Conference on Data Engineering (ICDE)*. Paris, France.

Lee, Y. J., Trevathan, J., Atkinson, I., & Read, W. (2015). The Integration, Analysis and Visualization of Sensor Data from Dispersed Wireless Sensor Network Systems Using the SWE Framework. *Journal of Telecommunications and Information Technology*, (4), 86.

OGC Standards, Open Geospatial Consortium (OGC), [Online, Accessed 05/01/2020]. Available: https://www.ogc.org /docs/is/.

Pradilla, J., Palau C., & Esteve, M. (2016). SOSLITE: Lightweight Sensor Observation Service (SOS) for the Internet of Things (IOT). *ITU Kaleidoscope: Trust in the Information Society*, Barcelona.

Sejdiu, B., Ismaili F., and Ahmedi L., (2020). "Integration of semantics into sensor data for the IoT - A Systematic Literature Review" - *International Journal on Semantic Web and Information Systems (IJSWIS)*. Volume 16, Issue 4, Article 1.

Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic sensor web. *IEEE Internet computing*, 12(4), (78-83).

W3C Semantic Sensor Network Incubator Group (SSN-XG), Semantic Sensor Network Ontology. [Online, Accessed 25/02/2020]. Available: https://www.w3.org /2005/ Incubator/ssn/ssnx/ssn.