IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams

T. Elsaleh[1], M. Bermudez-Edo[2], S. Enshaeifar[1], S. T. Acton[1], R. Rezvani[1], and P. Barnaghi[1]

[1]CVSSP, Department of Electrical and Electronic Engineering, University of Surrey, UK
[2]Software Engineering Department, University of Granada, Granada, Spain

***Abstract*—In recent years, the development and deployment of Internet of Things (IoT) devices has led to the generation of large volumes of real world data. Analytical models can be used to extract meaningful insights from this data. However, most of IoT data is not fully utilised, which is mainly due to interoperability issues and the difficulties to analyse data collected by heterogeneous resources. To overcome this heterogeneity, semantic technologies are used to create common models to share various data originated from heterogeneous sources. However, semantics add further overhead to data delivery, and the processing time to annotate the data with the model can increase the latency and complexity in publishing and querying the annotated data. In this paper, we present a lightweight semantic model to annotate IoT streams. The metadata descriptions that are provided in the models are used for search and discovery of the data using various attributes such as value and type. The proposed model extends commonly used ontologies such as W3C/OGC SSN ontology and its recent lightweight core, SOSA, and includes concepts to describe streaming IoT data. We also show use cases, tools and applications where the proposed model has been used.**

***Index Terms*—IoT, data model, ontology, data stream, semantic model.**

I. Introduction

The rapid growth of Internet of Things (IoT) devices has increased the amount of data that is collected and communicated from real world environments. The IoT data is often collected by various devices and is represented in different forms. The heterogeneity of the data and interoperability issues between different sources and platforms is a common challenge in creating large-scale IoT data analytics services and applications.

In order to solve the heterogeneity of the data, most of the IoT solutions are opting to apply semantics. Semantics provide common models to annotate heterogeneous data coming from heterogeneous sources of information. There are some semantic approaches that model IoT devices and some aspects about them, such as service offered, kind of data, units, etc. ([1], [2], [3], [4]). Recently some approaches have envisioned the need to provide a lightweight approach to the models, such as IoT-Lite [5], and the standard and popular Semantic Sensor Network model (SSN) [6], which recently has published a lightweight model with its core ontology, SOSA (Sensor, Observation, Sample, and Actuator) [7]. However, these models are centred around devices, but do not pay much attention to the IoT data streams. For stream annotation, some models have appeared that provide detailed annotations on stream data, such as SAO (Stream Annotation Ontology) [8]. However, at the time of stream data acquisition, some processing delay is added due to the detailed annotations. This restriction makes it complicated to deal with (quasi) real-time applications.

978-1-7281-2171-0/19/$31.00© 2019 IEEE

We propose a lightweight semantic model for the annotation of stream data, IoT-Stream, that extends SOSA, (and by extension SSN) ontology, and therefore is fully compatible with SOSA. In our proposal we have centred the attention around the stream data. To keep it simple, each stream observation contains only a value and a timestamp. To that end, we have taken outside the stream observation all the needed metadata useful for searching purposes, but unnecessary for the (quasi) real-time processing of the data. In our model, not only raw data can be annotated as streams, but also processed data. The processed data is also kept lightweight. For example, raw data coming from a sensor that it is produced every minute can be individually annotated. However, if such fine granularity is not needed, data can be aggregated in windows of 5 minutes using data mining algorithms, such as SAX (Symbolic Aggregate Approximation), and annotate the processed stream data. These aggregated stream annotations are still lightweight, with only the annotations of the aggregated value and the start/end times of the aggregated value. Metadata for searching purposes is also appended, such as algorithm used, values of the parameters of the algorithm, etc., outside of the stream observations. Towards the end of the paper some use cases, tools and applications where IoT-Stream has been successfully used are shown.

II. Related Work

Semantic models representing stream annotations are scarce. One representative is SAO. SAO has been built on top of some well-known ontologies to represent IoT data streams. `StreamData`, `StreamEvent`, `StreamAnalysis`, `Observation`, `Sensor` and `Segment` concepts enable this ontology to describe temporal concepts accurately. With `StreamData` class SAO can provide a stream data as a temporal point or segment and it describes the output of the observation as an event with `StreamEvent` class [8].

To represent IoT-Streams we need complementary concepts that represent devices, location, time and quantity units and values. In the descriptions of devices, there are some models to represent sensors and their observations. One of these models is the SSN ontology which describes sensors with their properties, systems, deployments, and stimuli and observations [6]. The SOSA ontology is a lightweight core for SSN that provides concepts for sensors, observation values and features of interest [7]. IoT-Lite is another lightweight model for IoT concepts with the aim of fast annotation, processing and semantic querying time. IoT-Lite was inspired by the IoT-A reference model [9] which defined core concepts for the IoT, namely Resources, Entities and Services. It instantiates and extends Device and Sensor concepts from SSN [5]. The focus of IoT-Lite, SOSA and SSN is more on sensing devices and is appropriate for sensor discovery.

There are some location models, such as Geo[1], that helps in searching for IoT devices. Geo is a popular model that represents location data in RDF, it does not try to tackle many of the matters covered in the professional Geographic Information System (GIS) world. Instead, the ontology offers just a few basic simple terms that can be used in RDF when there is a need to describe latitudes, longitudes and altitudes. The use of RDF as a carrier for latitude, longitude and altitude simplifies the capability for cross-domain data mixing, as well as describing entities that are positioned on the map (e.g. carrying out geospatial queries for Sensors, Deployments, Platforms or Systems). GeoSPARQL is a standard for the representation and querying of geospatial linked data for the Semantic Web from the Open Geospatial Consortium (OGC) [10]. GeoJSON[2] is a geospatial data interchange format based on JSON. It describes numerous types of JSON objects and the way they are joined to represent data about geographic features, their properties, and their spatial extents. GeoJSON supports a range of geometry types ranging from `Point`, `LineString`, `Polygon`, `MultiPoint`, `MultiLineString`, and `MultiPolygon`.

The Time ontology[3] is a well-known and widely used semantic model to represent time. It has vocabulary for representing information about topological (ordering) relations, duration and temporal position (i.e. date-time information). Time can be expressed using conventional clock, Unix-time, geologic time and other reference systems. For duration, it can also use different systems for example Gregorian calendar.

To provide quantities, units, dimensions and values there are also some ontologies. Qu ontology[4] is one of the well-known ontologies in this field. Qu ontology has been developed to support different Systems Modelling Language (SysML) users [11].

One important aspect is the Quality of Information (QoI), because faulty data can have costly consequences [12]. When talking about Quality of Information, categories or metrics are important to describe the details. There are five common metrics; Completeness, Correctness, Concordance, Currency and Plausibility [13]. In the CityPulse project[5], they extend the ontology and used five categories; Timeliness, Cost, Accuracy, Communication and Security, each with a bunch of sub-metrics. The major problem is lack of ground truth in Correctness which has been addressed in [14] with spatio-temporal, causality, and outcome evaluation.

III. IoT-Stream Ontology

The design of IoT-Stream is based on a set of principles, always keeping in mind that we aim for a lightweight ontology that extends SOSA to provide stream annotation concepts. In developing the model, we have followed several best practices found in the literature. First of all, the most followed guide for creating ontologies, created in 2001; ontology development 101 [15] should be followed. Secondly, in 2003, W3C published a list of sample Good Ontologies[6] following specific good practices, which has been considered here. And finally, in 2016, the development of the IoT-Lite ontology led to the extension of these guidelines to cover the scalability of ontologies.

The guidelines of ”ontology development 101”, divides the development of the ontology in 7 steps. In IoT-Stream the implementation of step 1 (Determine the domain and scope of the ontology) is the result of working extensively in many IoT projects and discussing requirements with several partners from data providers to application developers in different fields of IoT. This experience allows us to answer the questions of who would be the end user of the ontology, etc. Step 2 (Consider reusing existing ontologies) is derived from the study during several years of the ontologies in the area, and the selection of the relevant ones to be reused, or partially reuse. See for example section II. Steps 3 to 6 (Enumerate important terms in the ontology; Define the classes and the class hierarchy; Define the properties of classesslots; Define the facets of the slots) are defined further down in this section, and step 7 (Create instances) as illustrated in Figure 3 and its use described in section IV.

Similarly the ”Good Ontologies” list published by W3C scored the ontologies based on five aspects and in the development of IoT-Lite [5], authors published 10 steps for semantic model development. All these guidelines have been followed, and can be referred to in their respective references.

Following the previous guidelines, and keeping in mind always that the main principle is to provide a lightweight extension for data streams, we have developed IoT-Stream. The development of IoT-Stream has always followed the linked-data approach, that increases the chance of interoperability by extending popular ontologies. The SSN ontology has proved to have a significant impact in its adoption for semantically annotating IoT elements[7], and therefore, it is chosen as the ontology that IoT-Stream mainly extends, in addition to adopting its revised version, SOSA.

Hence, the aim of IoT-Stream ontology is to provide with the basic concepts needed to process stream data extending

[1] https://www.w3.org/2003/01/geo/

[2] https://tools.ietf.org/html/rfc7946

[3] https://www.w3.org/TR/2017/REC-owl-time-20171019/

[4] https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu-rec20.html

[5] http://www.ict-citypulse.eu/page/

[6] http://www.w3.org/wiki/Good_Ontologies

[7] https://w3c.github.io/sdw/ssn-usage/

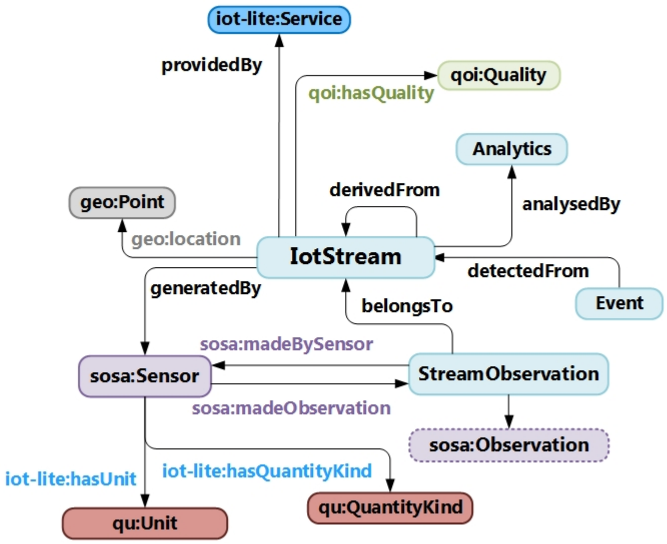


Fig. 1. IoT-Stream (in light blue) linked with the main ontologies

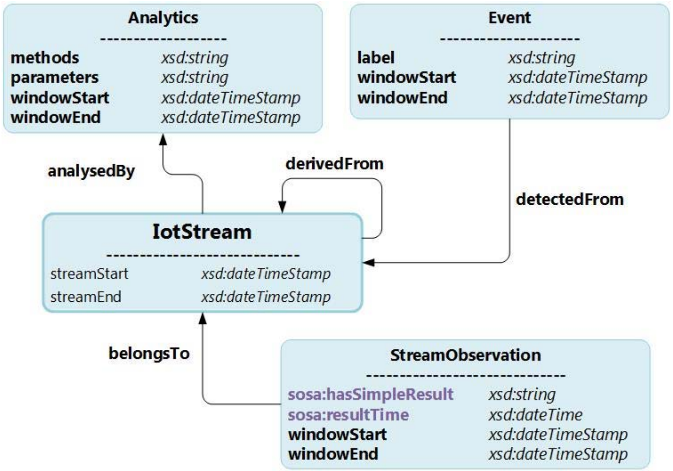

Analytics
methods xsd:string
parameters xsd:string
windowStart xsd:dateTimeStamp
windowEnd xsd:dateTimeStamp
Event
label xsd:string
windowStart xsd:dateTimeStamp
windowEnd xsd:dateTimeStamp
analysedBy
derivedFrom
detectedFrom
IotStream
streamStart xsd:dateTimeStamp
streamEnd xsd:dateTimeStamp
belongsTo
StreamObservation
sosa:hasSimpleResult xsd:string
sosa:resultTime xsd:dateTime
windowStart xsd:dateTimeStamp
windowEnd xsd:dateTimeStamp

Fig. 2. IoT-Stream classes and properties

SOSA in order to allow stream annotations. A huge percentage of the data retrieved from IoT devices is stored in streams. For example sensors measuring temperature, number of cars in a road, pollution, are recover as streams of data, with timestamps. When researches and developers need to process IoT data to obtain meaningful insights they have to deal with the heterogeneity of formats and syntactic of the data. To overcome this heterogeneity semantics can help. However, semantics in the early days was thought to give detailed annotations of the data, when there was no need for (quasi)real-time responses, and the amount of data was not huge. Therefore, the few semantic models that cover stream data, tend to describe the real world in detail with several concepts. One example of such semantic models is SAO, which was successfully used for forensics analysis and some (quasi)real-time analysis. However, when dealing with huge amounts of data, with high granularity it started to delay the analytics process. IoT-Stream intends to leverage the processing time of stream ontologies by annotating the streams with the minimum concepts needed to process the data.

IoT-Stream is composed only of 4 main concepts, as can be seen in Figure 1. These concepts are: `IotStream`, `StreamObservation`, `Analytics` and `Event`. The first concept created in the ontology was `StreamObservation`. This concept has only two temporal data properties, `windowStart` and `windowEnd`, and one temporal property `sosa:resultTime` from `sosa:Observation` (see Figure 2. Although the well-known Time Ontology provides temporal concepts, linking to these concepts for each `StreamObservation` would be too heavy. The other properties relating to the observation value are captured by `sosa:hasSimpleResult`. The rest of the concepts with other useful information are kept outside this concept. With this measure, the retrieval of stream values, once the source is selected, is just providing values and timestamps, keeping the queries and responses light. From this concept, we have created another concept, the `IotStream`, which links all `streamObservations` that `belongsTo` a specific stream. `IoTStreams` can also be `derivedFrom` other `IoTStreams` which could be the result of being `analysedBy` data analysis techniques. The techniques applied on the stream are captured in the `Analytics` class, in which the `methods` and their corresponding `parameters` are defined as a vector string. For example, we can have a stream of raw data and apply an aggregation algorithm to increment the granularity of the data (see examples in the next sections). This concept is not only for aggregation but for any other analytics that convert an stream into another stream. Using analytics, `Events` can be `detectedFrom` single or multiple stream. For example if it is a rainy or sunny day, derived from humidity and temperature streams.

One of the main contributions of the ontologies is the power to share and link data from different models. That is why most of the semantic community has adopted the linked approach, and now we can find hundreds of semantic models linked together[8], that can share their annotated data, without further conversion efforts. Regarding the linked-data we have, the literature and semantic models published have been studied (see section II). Then, a set of ontologies have been selected to use together with IoT-Stream (see the ontologies described here and used in the examples in the next sections). Only stable and well-known ontologies have been chosen, generally with a strong body behind them, such as W3 or Open Geospatial Consortium (OGC). This is to ensure that the ontologies are dereferenceable and that they will not disappear, at least in the near future. For example, for the annotation of sensor devices and services we have used SOSA [7] and IoT-Lite [5] ontologies. Normally, IoT applications need to also search by location and quantity kind, etc. Therefore, the ontology has been linked to several existing ontologies that describe these concepts. For location, the Geo ontology is used, and

[8] https://lod-cloud.net/versions/2017-02-20/lod.svg

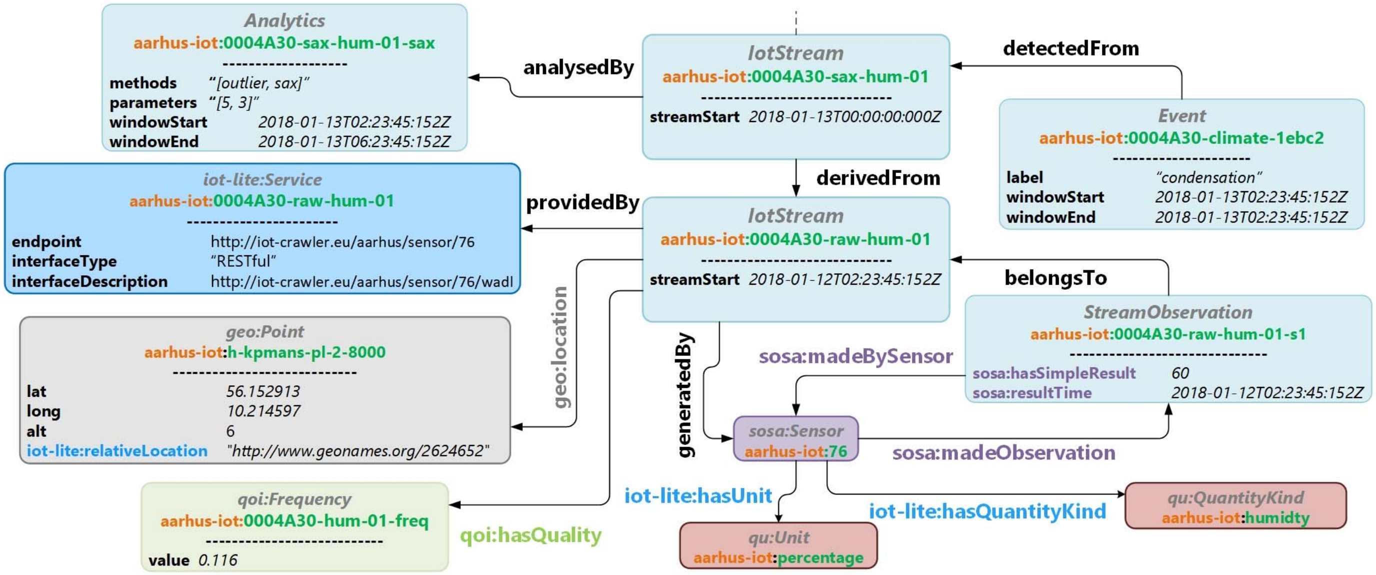

Analytics
aarhus-iot:0004A30-sax-hum-01-sax
methods "[outlier, sax]"
parameters "[5, 3]"
windowStart 2018-01-13T02:23:45:152Z
windowEnd 2018-01-13T06:23:45:152Z
analysedBy
IotStream
aarhus-iot:0004A30-sax-hum-01
streamStart 2018-01-13T00:00:00:000Z
detectedFrom
Event
aarhus-iot:0004A30-climate-1ebc2
label "condensation"
windowStart 2018-01-13T02:23:45:152Z
windowEnd 2018-01-13T02:23:45:152Z
iot-lite:Service
aarhus-iot:0004A30-raw-hum-01
endpoint http://iot-crawler.eu/aarhus/sensor/76
interfaceType "RESTful"
interfaceDescription http://iot-crawler.eu/aarhus/sensor/76/wadl
providedBy
derivedFrom
IotStream
aarhus-iot:0004A30-raw-hum-01
streamStart 2018-01-12T02:23:45:152Z
belongsTo
StreamObservation
aarhus-iot:0004A30-raw-hum-01-s1
sosa:hasSimpleResult 60
sosa:resultTime 2018-01-12T02:23:45:152Z
geo:location
geo:Point
aarhus-iot:h-kpmans-pl-2-8000
lat 56.152913
long 10.214597
alt 6
iot-lite:relativeLocation "http://www.geonames.org/2624652"
generatedBy
sosa:madeBySensor
sosa:Sensor
aarhus-iot:76
sosa:madeObservation
qoi:Frequency
aarhus-iot:0004A30-hum-01-freq
value 0.116
qoi:hasQuality
iot-lite:hasUnit
qu:Unit
aarhus-iot:percentage
iot-lite:hasQuantityKind
qu:QuantityKind
aarhus-iot:humidty

Fig. 3. An instance of an IoT-Stream

for observation coverage, IoT-Lite and GeoSPARQL [10] are employed. Quantity kind and unit taxonomies come from QU-rec [11]. Taking this approach, `IoTStreams` link to concepts from other ontologies to capture information about the `qoi:Quality` of the stream, the `iot-lite:Service` its `providedBy`. It also links to the `sosa:Sensor` that it was `generatedBy`, the `qu:QuanityKind` and `qu:Unit` that the `sosa:Sensor` measures with, and also the `geo:Point` where the stream originates from.

Therefore, upon querying, we have centred the ontology around the `IotStream` concept, providing from this concept direct links to the needed information to form common searches. Having a centred concept, the queries become lighter, because they need less triples to find each aspect of the search.

IV. Use Cases

An illustrative approach to demonstrate how IoT-Stream can be used is to apply it in use cases.

A. Smart Healthy Living

In the ACTIVAGE project [16], a large-scale pilot is being conducted in different deployment sites within Europe. The pilot mainly focuses on the provision of an IoT platform for promoting active and health living, particularly for older people. The project has defined several use cases to achieve this. Several use cases such as daily activity monitoring involve the application of smart technologies in the home environment, by deploying a range of sensor devices[9] that capture physiological observations such as step count, heart rate, ambient phenomena such as temperature and humidity, and general activity in the home such motion, energy consumption of certain appliances (See Figure 4). In such scenario, data needs to be sent continuously to the platform for storage and analysis based on the use cases applied to the home environment. Here, metadata about the data stream can be annotated using IoT-Stream and the ontologies that it links to. Sensor identification, phenomena, and units are captured by SSN/SOSA. Data streams, observations, applied analysis techniques, and events for emergency triggers, such as falls or persistent air quality degradation, can be captured by IoT-Stream. Another critical aspect with regards to pilot deployments is the capturing of the quality of information for data streams with respect to consistency and reliability, which is captured by QoI [17]. For a platform that is distributed among different remote sites, the discovery of data stream services from different sites captured by IoT-lite would allow for analysis studies such as cross-correlation between users,

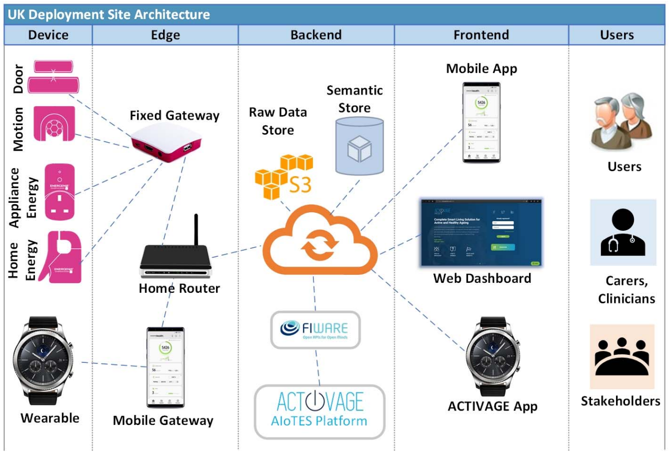
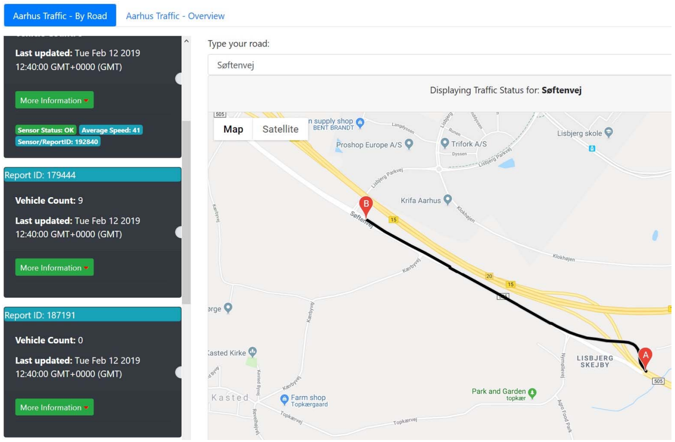


Fig. 4. Capturing IoT Stream data from Smart Home environments

[9] http://www.activageproject.eu/deployment-sites/Leeds/



Last updated: Tue Feb 12 2019 12:40:00 GMT+0000 (GMT)

Sensor Status: OK Average Speed: 41 Sensor/ReportID: 192840

Report ID: 179444

Vehicle Count: 9

Last updated: Tue Feb 12 2019 12:40:00 GMT+0000 (GMT)

Report ID: 187191

Vehicle Count: 0

Last updated: Tue Feb 12 2019 12:40:00 GMT+0000 (GMT)

Type your road: Søftenvej

Displaying Traffic Status for: **Søftenvej**

Fig. 5. Smart City Traffic Analysis

groups or services.

B. *Smart City Traffic*

Open data banks often provide a wide range of rich datasets with metadata, as well as near real-time IoT `streamObservations` from different parts of the world. The city of Aarhus open data bank provides real-time data and metadata for traffic measurement using IoT devices deployed across the city monitoring approximately 400 points[10] on selected areas. The smart sensors are able to detect and register active Bluetooth devices (mobiles phones) from moving motor vehicles. By measuring the times it takes for a motor vehicle to travel from its starting detection point to the end detection point, the calculated speed can be converted to a map showing the roads current accessibility. This rich data stream can be annotated by IoT-Stream and its linked ontologies. The IoTStream `Analytics` class can both be used in relation to traffic analysis and to observe the day to day developments such as roadworks. The `Event` class can capture the most important (irregular) road events such as low or high traffic within the municipality. Knowing the deployed location of the sensors within the municipality is also vital for geospatial analysis, so with IoT-Stream linking to `geo:Point`, this can be captured using absolute co-ordinates or relative location (using iot-lite:relativeLocation) (See Figure 5).

V. Tools and Applications

A. *Data Analysis Tools*

Data analysis plays a very important role for IoT streams, as the nature of these streams can be potentially massive, erroneous, and intermittent. Therefore, techniques such as data pre-processing, machine learning and correlation are needed for aggregation, filtering, re-sampling and pattern detection. The Knowledge Acquisition Toolkit (KAT) service [18] is a web application that exposes a suite of methods that implement these techniques. The application can be deployed either on a local machine with a graphical user interface, or on a server exposed with a RESTful interface. KAT is able to retrieve a dataset from a semantic data broker by employing SPARQL templates that format the result in a manner that can be processed by KAT. The output of the service will be a new data stream based on the methods that were used, and their corresponding parameters. The dataset is then re-annotated to a new ”analysed” IoT stream whereby the details of the analysis process are declared in the `Analytics` class (See Figure 6).

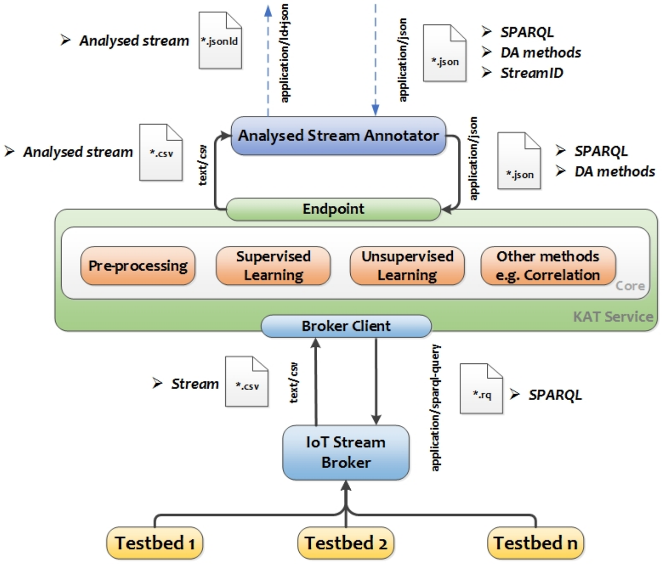


Fig. 6. Analysed IoT Stream using KAT service

B. *Crawling and Search Engines for IoT Data Streams*

Another application for IoT Stream is demonstrated in the IoTCrawler [19] framework. A crawler and search engine prototype for IoT data stream sources has been designed and developed which uses information annotated using IoT Stream and the ontologies linked to it more extensively for the integration of IoT data from different domains. The crawler extracts metadata from the data sources and stores them in a metadata repository. The data consists of an identifier for an `IoTStream`, the identifier of the `sosa:Sensor` generating it, the corresponding `qu:Unit` and `qu:QuantityKind`, the `geo:location` and the `iot-lite:Service` that provides the stream data. Several processing layer components discover data stream providers from the repository, and then invoke their brokers for their data streams. The processing components then extract insights by applying data analysis, such as the measurement of the Quality of Information `qoi:Quality` and also the detection of Events `iot-stream:Event`. Then through a process termed as semantic enrichment, the new insights are then annotated and then stored into the metadata repository. The search engine then accesses the metadata repository for indexing the data stream services based on location (using geohashing), themes, events and quality of information. On the client side, users and 3rd party applications are able to perform instant searches

[10] https://portal.opendata.dk/dataset/realtids-trafikdata

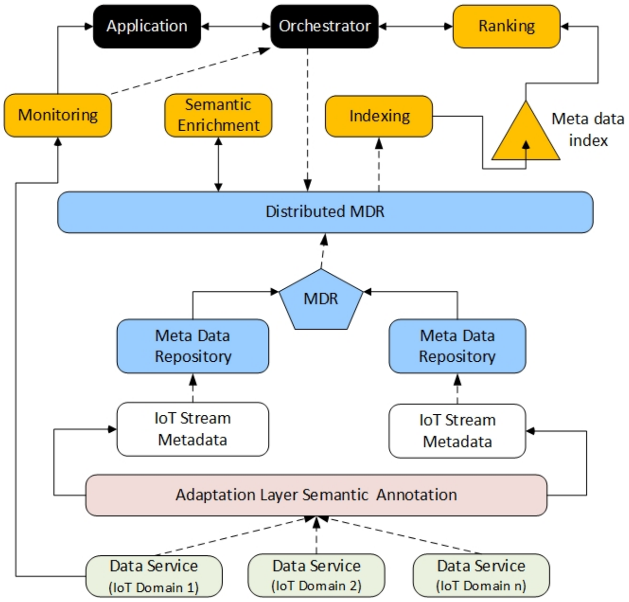


Fig. 7. IoTCrawler Architecture with use of IoT-Stream

or subscribes to updates data stream services based on their preferences (See Figure 7).

VI. Conclusions

In this paper we presented IoT-Stream, a novel semantic model for stream annotations. We have developed the model according to the most recognised and state-of-the-art guidelines to develop semantic models, and specially guidelines for IoT environments, where the scalability and short processing time are essential. With these restrictions, we have created a lightweight semantic model fully compatible and as an extension of the well-known SSN ontology and its recent lightweight core SOSA. With the main concept, `StreamObservation`, with simple temporal and value properties, we have accelerated the queries to the stream model and have created the rest of the concepts needed for crawling and searching streams around this main concept. By doing so, we have improved the processing time of the stream queries. We have proved the validity of IoT-Stream through a series of real annotation scenarios. The ontology is publicly available[11] with dereferenceable concepts and available in several formats, some of them from the field of ontologies, such as RDF and Turtle, and some of them from the field of Web development, such as JSON-LD (which are the fields of the potential developers using the ontology). This variety of formats will facilitate the adoption of the ontology from different developers with different backgrounds, and heterogeneous applications, making interoperability easier. For fast adoption of the model, and even for those not experts on ontologies, we have developed several accompanied tools, such as annotators, shown in the use cases, stream analysers (such as KAT) and crawling and searching engines.

Acknowledgement

This work has been sponsored by the EU Horizon 2020 Research and Innovation programme, through projects IoTCrawler (contract no. 779852) and ACTIVAGE (contract no. 732679)

References

[1] Ş. Kolozali, M. Bermudez-Edo *et al.*, "Observing the pulse of a city: A smart city framework for real-time discovery, federation, and aggregation of data streams," *IEEE Internet of Things Journal*, 2018.
[2] N. Seydoux, K. Drira *et al.*, "Iot-o, a core-domain iot ontology to represent connected devices networks," in *European Knowledge Acquisition Workshop*. Springer, 2016, pp. 561–576.
[3] M. Alirezaie, K. Hammar *et al.*, "Smartenv ontology in e-care@ home," in *9th International Semantic Sensor Networks Workshop*, 2018, pp. 72–79.
[4] E. G. Petrakis, S. Sotiriadis *et al.*, "Internet of things as a service (itaas): Challenges and solutions for management of sensor data on the cloud and the fog," *Internet of Things*, 2018.
[5] M. Bermudez-Edo, T. Elsaleh *et al.*, "Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 475–487, 2017.
[6] M. Compton, P. Barnaghi *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.
[7] K. Janowicz, A. Haller *et al.*, "Sosa: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, 2018.
[8] S. Kolozali, M. Bermudez-Edo *et al.*, "A knowledge-based approach for real-time iot data stream annotation and processing," in *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE*. IEEE, 2014, pp. 215–222.
[9] A. Bassi, M. Bauer *et al.*, *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*, 1st ed. Springer Publishing Company, Incorporated, 2016.
[10] R. Battle and D. Kolas, "Enabling the geospatial semantic web with parliament and geosparql," *Semantic Web*, vol. 3, no. 4, pp. 355–370, 2012.
[11] O. Specification, "Omg system modeling language (omg sysmltm) specification," 2006.
[12] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data quality in context," *Communications of the ACM*, vol. 40, no. 5, pp. 103–110, 1997.
[13] N. G. Weiskopf and C. Weng, "Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 144–151, 2013.
[14] R. Zafarani and H. Liu, "Evaluation without ground truth in social media research," *Communications of the ACM*, vol. 58, no. 6, pp. 54–60, 2015.
[15] F. N. Natalya, L. Deborah *et al.*, "Ontology development 101: A guide to creating your first ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, 2001.
[16] European project ACTIVAGE. [Online]. Available: https://www.activageproject.eu/
[17] T. Iggena, D. Kümper, and R. Tönjes, "Kontinuierliche bewertung von informationsqualität in stream-basierten smart city architekturen," *ITG-Fachbericht-Mobilkommunikation–Technologien und Anwendungen*, 2014.
[18] F. Ganz, D. Puschmann *et al.*, "A practical evaluation of information processing and abstraction techniques for the internet of things," *IEEE Internet of Things journal*, vol. 2, no. 4, pp. 340–354, 2015.
[19] European project IoTCrawler. [Online]. Available: https://www.iotcrawler.eu/

[11] http://iot.ee.surrey.ac.uk/iot-crawler/ontology/iot-stream