A Knowledge-based Approach for Real-Time IoT Data Stream Annotation and Processing

Şefki Kolozali, Maria Bermudez-Edo, Daniel Puschmann, Frieder Ganz, Payam Barnaghi
Centre for Communication Systems Research
University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom
{*s.kolozali, m.bermudez, d.puschmann, f.ganz, p.barnaghi*}*@surrey.ac.uk*

***Abstract*—Internet of Things is a generic term that refers to interconnection of real-world services which are provided by smart objects and sensors that enable interaction with the physical world. Cities are also evolving into large interconnected ecosystems in an effort to improve sustainability and operational efficiency of the city services and infrastructure. However, it is often difficult to perform real-time analysis of large amount of heterogeneous data and sensory information that are provided by various sources. This paper describes a framework for real-time semantic annotation of streaming IoT data to support dynamic integration into the Web using the Advanced Message Queuing Protocol (AMPQ). This will enable delivery of large volume of data that can influence the performance of the smart city systems that use IoT data. We present an information model to represent summarisation and reliability of stream data. The framework is evaluated with the data size and average exchanged message time using summarised and raw sensor data. Based on a statistical analysis, a detailed comparison between various sensor points is made to investigate the memory and computational cost for the stream annotation framework.**

***Keywords*-Internet of Things; Smart Cities; Semantic Stream Annotation; Linked Sensor Data; Knowledge Management.**

I. Introduction

Recent advances in Information Communication Technologies (ICT) have led to a technological turn in an increasing number of cities by enabling a new type of embedded spatial intelligence that advances the information and knowledge capabilities of communities. The Internet of Things refers to interconnection of diverse objects or Things that allows to monitor the physical world and provide their virtual representations on the internet. The availability of such connected objects and sensors open opportunities to observe the status of the physical world in real time, and process this information to improve the operational efficiency of the city services and infrastructure.

The integration of a large amount of multi-modal data streams from diverse application domains (e.g. traffic information, parking spaces, bus timetables, waiting times at events, event calendars, environment sensors for pollution or weather warnings, GIS databases) is one of the key challenges in developing smart city frameworks. Therefore, knowledge management is a primary concern for smart cities, and until recently most of the solutions that have been created for various scenarios and applications were existing in isolation. For instance, traffic or governmental data, encyclopedic sources such as Wikipedia[1], and forecast data collection (e.g. the UK Met Office[2]) do not interact with each other, even though they can deal with similar kind of data. As a result, information managed by one of these resources may not benefit from information held by the other. This issue becomes much more noticeable once we consider the exchange of results between IoT researchers. For this reason, linkage of the results with other data sources to enable them to be meaningful is as much essential as providing access to IoT data stream through web services.

The quality of information and services is another challenge that can influence the usability of a data source in heterogeneous and distributed environments. In smart cities, data sources that are emerged from diverse application domains can have different qualities, modalities, and trust and reputation, which need to be identified and associated to a set of criteria that represent the quality of information and service. Given that cities are dynamic and evolving eco-systems, there is also a need to continuously link, interpret and share dynamic knowledge across city stakeholders and citizens in order to utilise information before it is outdated. Real-time stream annotation is, therefore, another critical endeavour that ought to be dealt within smart city frameworks.

To overcome these issues within the domain of smart cities, we propose a framework developed in the scope of the CityPulse project[3] for real-time IoT stream annotation that employs a knowledge-based approach to represent data streams and to support mashups. To deal with large amount of data, we use Advanced Message Queuing Protocol (AMPQ) as proposed in [12] to increase the performance of the system. We also present an information model to provide a representation for summarisation and reliability of IoT stream data. In order to investigate the performance of the framework, a traffic dataset is collected from a city environment. The framework is evaluated with the data size

[1]https://www.wikipedia.com
[2]http://www.metoffice.gov.uk/
[3]http://www.ict-citypulse.eu/page/

and average exchanged message time using summarised and raw sensor data to investigate the memory and computational cost for the stream annotation framework. This work is based on our previous work [1] that offers a way to represent data stream, and enriches it with semantic annotations. The remainder of the paper is organised as follows. Section II describes the related work. Section III details the overall functional components of the proposed smart city framework. Section IV demonstrates the proposed framework for semantic annotation of streams and presents an information model to express summarisation and reliability of stream data. Section V provides a use case scenario that illustrates the semantic annotation of a stream data in our system. Section VI details an evaluation of the proposed framework and Section VII concludes the paper and describes the future work.

II. Related Work

IoT research in recent years has focused on modelling domain knowledge of sensor networks and services. The SSN ontology [3] is one of the most significant efforts in development of an information model for sensory data. The SSN Ontology provides a vocabulary for describing concepts such as sensors, outputs, observation value and feature of interests. Most notable extensions include ontologies for coastal features, services and roles for emergency response. However, although the SSN ontology defines a high-level scheme of sensor systems, it does not include representation of observation and measurement data. IoT-A model [4] and IoT.est semantic representations [13] describe how to enhance the utilisation and representation of domain knowledge in sensor networks where the former provided an architectural base for further IoT projects, and the latter enhanced the IoT-A model with some service and test concepts.

The Observation and Measurement (O&M) descriptions for sensory data are also described as a part of the Sensor Web Enablement (SWE) standards [2] from the Open Geospatial Consortium (OGC). While it provides several important syntactic descriptions, due to the fact that it is based on XML, it has a weak semantic structure for expressing knowledge, and lacks some important features to describe an ontology in more detail. There has been a recent study to improve the semantic richness of the O&M ontology where authors transformed it into Ontology Web Language (OWL) representation [6]. However, the O&M ontology continues to not only lack temporal features to represent time-series observations in detail, but also semantics for our purposes due to the straightforward approach that has been used in the process.

Another very similar approach has been carried out in [7], in which all the XML tags of O&M ontology have been mapped into OWL concepts. However, although the authors present how to access the annotated data through SPARQL queries, these queries are not efficient for the applications that need to access sensor data in real time, as the SPARQL queries generates significant traffic if the sampling rate is small. Consequently, there still remains a need for a framework to handle real-time semantic annotation as well as efficient knowledge representation of sensory data in dynamic environments such as smart cities.

III. Smart City Framework

The CityPulse framework aims to provide an infrastructure to address the complex task of stream processing by providing large-scale data analysis and real-time intelligence functionalities. Figure 1 illustrates an overview of the CityPulse framework.

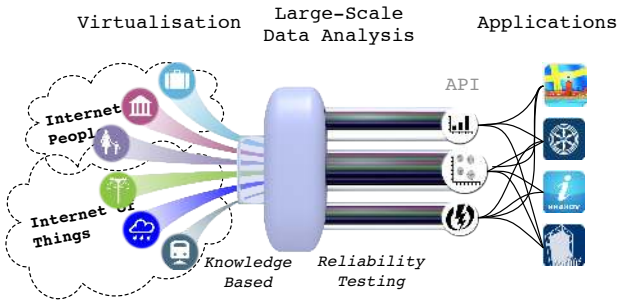


Figure 1: An overview of the different data sources and key areas involved in the CityPulse framework.

Sensory data streams involve rapid changes due to dynamicity of their environment and are employed on resource constraint platforms. Processing and detecting an event is a more challenging task compare to the conventional stream data. Therefore, using energy efficient methods as well as multi-granular representation and management of IoT streams is a challenging task. The CityPulse project provides energy and process efficient solution combining data aggregation and pattern creation to respond to real-time requests considering resource limitations of devices that provide IoT data.

Smart cities operate in dynamic environments in which the properties of underlying services and resources dynamically change and depend on physical world events and phenomena (e.g. sensor readings - network availability, weather conditions, and temperature). We utilise a domain knowledge to interpret the aggregated data streams and detect higher-level events (i.e. machine interpretable or human understandable events) from Cyber-Physical-Social streams. We also aim to supply a user-centric decision support which makes use of contextual information, usage patterns and preferences to offer ideal configurations of smart city applications. This enables users (e.g. citizens, enterprises or city councils) to explicitly specify the requirements and personal preferences and also finds related information in line with the users'

preferences and context dependent attributes (e.g. location, time) through a matchmaking mechanism.

IV. Real-Time Stream Annotation

The real-time annotation framework aims to semantic annotation of IoT stream data by taking into account dimensionality reduction and reliability. The framework involves four main units: a) virtualisation, b) middleware, c) reliable information processing and d) semantic annotation. Figure 2 depicts the architecture of the framework.

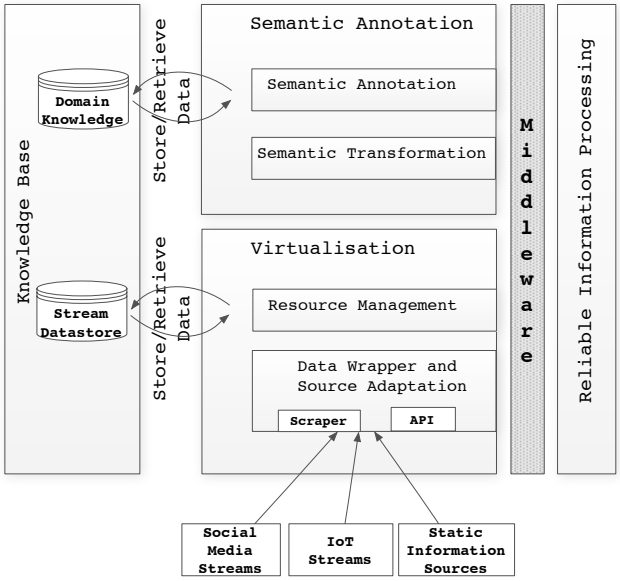


Figure 2: Framework for real-time stream annotation in smart city applications

A. Virtualisation

The virtualisation component facilitates access to heterogeneous data sources and infrastructure concealing the technical facets of data streams such as location, storage structure, access format, and streaming technology. The system designates various wrappers to encompass a large number of input formats, while it provides a unified format as output which is defined in the system.

In the context of smart cities and real-time information processing, the IoT resources represent the city sensors and actuators as well as data repositories which collect the information relevant to the operation of the city. The IoT Resource virtualisation allows modelling of the resources (e.g. sensors, actuators, data repositories, citizens) in a manner which enables a device, such as a parking application, to access these resources systematically. The citizen communication devices (e.g. smart phones) can be also used as virtual sensors with city experience to be the observed phenomena.

B. Middleware

There are many solutions that offer communication in distributed systems. The shortcomings of the alternatives are combinations of coupling of space (i.e. sender and receiver need to hold references about each other), time (i.e. the components need to interact at the same time) and synchronisation (i.e. the individual components block their activity while waiting for other processes to finish). To solve these issues we use a publish/subscribe mechanism which decouples time, space and synchrinasation. Furthermore the message delivery logic is handled via a message broker, decoupling it from the application layer.

In particular we are using the Advanced Message Queue Protocol (AMQP) which has been introduced in [12] as an open standard for message oriented middleware. The protocol divides the message brokering task into exchanges and message queues, whereby the exchange decides which messages will be pushed into which queue. This leads to enhanced flexibility for developers and avoids the need for static implementations. In order to handle scalability issues which arise in the context of IoT and Smart City data, we propose aggregating the data before passing it through the middleware. This phenomenon was demonstrated in [5], where it has been pointed out that the data abstractions can help to reduce data traffic and ultimately even energy consumption at the sensory level. The messages in our system have three fields. The first field is "message types": it defines the type of message. The second field is "metadata": it contains location and time information as well as information about the data source. The third field is "data": it consists of the raw values and identifier.

We have defined three types of messages: *transform*, *store* and *forward*. Generally the messages include all the three types. In our case, the subscriber can perform some computations on the data, or store it for later evaluations and then publish the transformed data. For instance, a subscriber can add semantic annotations to the data, while another one performs Quality of Information (QoI) computation. This approach allows different components to work asynchronously on stream data. The semantic annotations can be instantly accomplished, as the QoI measurements will simply be delivered when the corresponding data has been collected after a certain amount of time (e.g. a month). Following this time period, the affiliated data will be published in the message queue once again to ensure the semantic data store can update this missing QoI values.

C. Reliable Information Processing

The dynamicity and heterogeneity of IoT environments involves changes and prone to errors in the data, specially when dealing with crowd sourced data. The methods for information extraction and data processing, however, require accuracy and trust issues to be taken into account. This module measures and process accuracy and trust in data

acquisition, federation and aggregation. It integrates techniques for monitoring and testing, ensuring reliable information processing. For example, it provides fault tolerance mechanisms when malfunctioning or disappearing sensor are detected, or providing conflict resolution strategies when data analysis result in conflicting information. Provenance also plays an important role in Smart cities applications. These applications acquire data from heterogeneous sources, some of them more reliable (e.g. government data), and some of them less reliable (e.g. crowd sourced data). Based on user or application preferences, the application provider could choose to use less reliable data in cases that it has more up to date information. The reliable information processing module performs provenance analysis to assert the reliability of the data.

D. Data Modelling and Semantic Annotation

Smart city applications use data from different stream sources. Therefore the amount of traffic generated by these applications can be voluminous, particularly for real time applications in environments with resource constrains devices, for example sensors with limited bandwidth, memory or power. On the one hand, the proposed data model should be lightweight in order to reduce the traffic and processing time. On the other hand, it should explicitly represent the meaning and relationships of terms in vocabularies. In this study, we present a lightweight data model, which uses well-known models to represent IoT. The ontology contains 3 main modules, namely Stream Annotation Ontology (SAO), Quality of Service and Quality of Information (QoS|QoI), and provenance. Figure 3 shows an overview of the proposed information model. Some of this modules has been adapted from the IoT.est model [13]. In the next subsections we describe these three modules.

1) SAO module: Representing IoT stream data is an important requirement in semantic stream data applications, as well as in knowledge-based environments for Smart Cities. The SAO can be used to express the features of a stream data. It allows publishing content-derived data about IoT streams and provides concepts such as `sao:StreamData`, `sao:Segment`, `sao:SegmentAnalysis` on top of the TimeLine[4][11] and IoTest models. The SAO uses the broad definition of the `StreamEvent` concept in order to express an artificial classification of a time region, corresponding to a particular stream data. It also extends the sensor observations described in SSN Ontology (`ssn:Observations`) through a concept, `sao:StreamData`, that allows to describe `sao:Segment` or `sao:Point` linked to time intervals or time instants. Figure 4 shows the basic structure of the ontology.

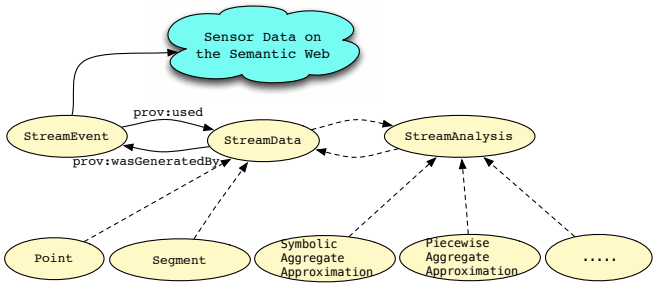


Figure 4: Depiction of the main concepts and relationships in the Stream Annotation Ontology.

In the context of smart cities, dimensionality reduction of data stream or stream transformations obtained through shifted (overlapping) windows can results in a data rate, different from the sample rate of the original sensor observation. Using the SAO Ontology, we can describe a data stream and a timeline instance to link the segment description with the time extent of a temporal entity representing the data stream. Thus, we can express a stream data as a time interval on the universal timeline, and also relate such an interval with the corresponding interval on the discrete timeline along with its discrete sampling rate. With regards to the previous conceptualisations of sensory data, the SAO ontology deals with representation of aggregated stream data and temporal characteristics. It is free from deep taxonomical organisation, and does not attempt to describe the deep interrelationships or computation of stream data.

2) QoS and QoI module: Quality of Service (QoS) has been widely studied in sensor networks, and has well defined and measurable properties ,such as throughput, jitter or packet loss, inherited from the field of network communications. However, although it has been spotted as one crucial item in data networks, the Quality of Information (QoI) is still not well defined and sometimes difficult to measure. In our model we have designed the QoI module based on the IoT.est model, and enhanced it with some related concepts, as well as with the help of experts in the field of data networks and data applications. Figure 5 depicts some of the concepts that are included in the QoI module. Some of these concepts could be directly annotated from the raw stream data and others need data analysis to be quantified. The process is performed in the reliable information processing component of the framework shown in Figure 2.

3) Provenance module: Provenance annotation helps in tracking the source of information and evaluates trustworthiness of different sources of information. Provenance can also track the algorithms, sample rate and other useful processing properties. These annotations specifies the reliability of the information and the adequacy of the data for a particular

[4]Timeline Ontology extends OWL-Time with various timelines (e.g. universal or discrete), temporal concepts, such as instants and intervals, and interval relationships. Available at: http://motools.sourceforge.net/timeline/timeline.html

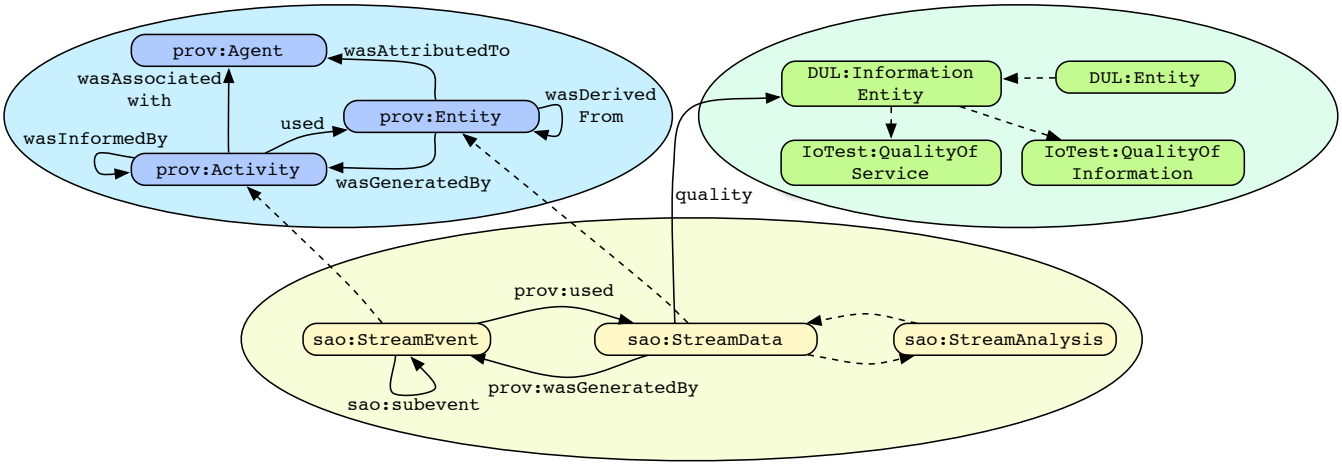


Figure 3: Describing a stream annotation work flow using the Stream Annotation Ontology.

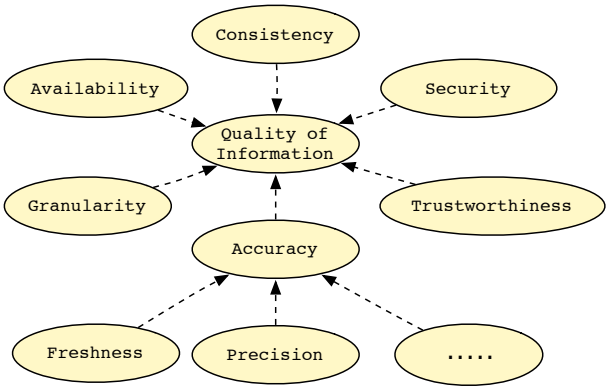


Figure 5: An overview of the Quality of Information module.

application. In our data model the provenance module only contains a few object properties that link the SAO module with the prov-o ontology[5].

V. Use Cases

One of the key issues in heterogeneous ecosystem of smart cities is real-time traffic data analysis. Enabling smart cities to efficiently manage traffic data and provide alternative routes will not only help in reducing transportation cost but also pollution that has been caused by traffic congestion. As a use case scenario, we use public traffic data[6] that has been obtained from the city of Aarhus in Denmark. The database consists of traffic data that has been measured among various sensors in different cities providing information regarding the geographical location, timestamp, and traffic intensity such as average speed and vehicle count. The data is taken from 135 sensors and samples every 5 minutes.

[5]http://www.w3.org/TR/prov-o/

[6]http://www.odaa.dk/dataset/realtids-trafikdata

While most of the systems constantly create and transmit raw sensor data, we need to be able to express a narrower, more specific workflow such as representation of aggregated and summarised data for individual sensor recordings and the smart city workflow. This will pave the way towards having scalable systems, and reduce memory and computational cost of massive amount of real-time data produced by sensors. For this reason, the semantic representation of the summarised data is as important as annotation of raw stream data. In this section, we exemplify the use of the proposed information model, describing the outcome of a pair of sensor recordings, and its representation in a road traffic environment.

Figure 6: A visual representation of geographical coordinates on Google Map for a pair of road traffic sensors provided by city of Aarhus, Denmark.

Figure 6 illustrates a sample location from city of Aarhus in Denmark on Google Maps showing a pair of traffic sensor points, that have been virtualised in our system, and aggregated and semantically annotated based on Stream Annotation Ontology. The sensor points refer to exact geographical coordinates (i.e. latitude, longitude), and linked

to resources such as DBPedia[7] and GeoNames[8] that are publicly available as a part of the Linked Open Data cloud. In aggregation process, the streaming data that has been obtained from these sensors, divided into segments and a patterns is created for each segment. These patterns represent an aggregation of a set of raw sensor data during a period of time. The pattern construction is performed using the Symbolic Aggregate Approximation (SAX) technique [9]. SAX is used in data mining and time series data for dimensionality reduction and creation of symbolic patterns. It divides a time series data into equal segments and then creates a string representation for each segment. Figure 7 depicts the data captured for average speed via the corresponding sensor points and illustrate SAX patterns created from the raw data.

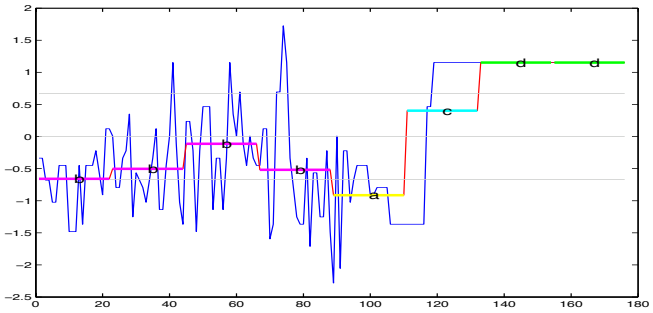


Figure 7: A real time average speed data obtained from a pair of sensor points given in Figure 6 is mapped into SAX word, "bbbbacdd", with the segment size of "8" and alphabet size of "4" for 176 samples.

In listing 1, we describe a set of sensor recordings obtained from the sensor platforms, given in Figure 6, and represent summarised data, shown in Figure 7, as well as temporal entities using the Stream Annotation Ontology. As the proposed semantic model is directly connected to the PROV-O Ontology, we can track the provenance of the information. For instance, in this case the raw data is coming from a public provider, and it has been processed with the stream analysis algorithm SAX, then it has been stored as a stream observation in SAO ontology. This provenance tracking can be used to measure the reliability of the information. With the reliability results, the application developer or the user can make the decision to trust the information or not. We can also annotate QoI concepts, sch as `freshness` of the data, taken from the database field *timestamp*; `availability`, taken from the database field *status*; `granularity`, taken from the database field *VehicleCount*.

VI. Evaluation

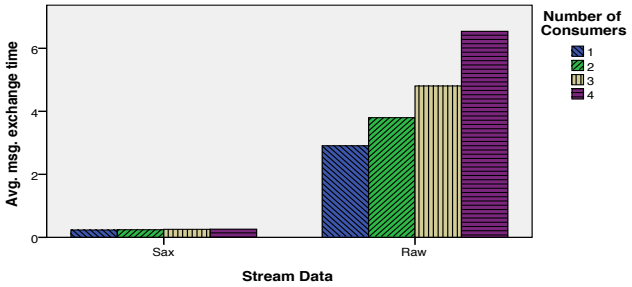
In this section, the performance of the stream annotation framework is evaluated using data size and average message exchange time. The evaluations were performed using a RabbitMQ server, that is based on AMQP, on a Personal Computer (PC) running Windows 7 Professional operating system with an Intel Core i7-2670QM 2.2GHz processor and 4GB RAM memory. The aim of this experiment is to send the raw and summarised data as messages through middleware with a different number of consumers to read the messages.

Our experimental dataset consists of two sets of stream samples: (i) raw dataset that contains 72240 samples, and (ii) summarised dataset that contains 444 samples of the sensor stream data obtained from the road traffic of the city of Aarhus. The overall results of average message delivery time are obtained by averaging the results obtained in the 10 experimental runs. The level of accuracy estimated by these metrics were analysed utilising a two-way Multivariate Analysis of Variance (MANOVA). The independent variables were the data dimension (i.e. raw and sax stream data), and number of consumers (i.e. 1, 2, 3, 4). The dependent variables were the following metrics: the size of data in KiloBytes (KB), and the average message exchange time in second. The Holm-Sidak procedure [8] and a risk α of .05 were used in the MANOVA tests. In addition, we have used the following definitions in our interpretations of the effect sizes: small effect size ($\eta^2 \leq .01$), medium effect size ($.01 \leq \eta^2 \leq .06$) and large effect size ($.06 \leq \eta^2 \leq .14$).

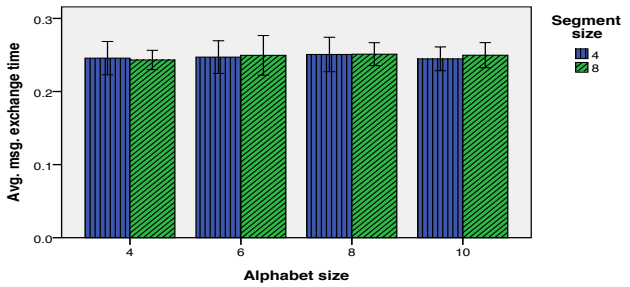
```
@prefix sao: <http://example.com#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
@prefix qoi: <http://example.com/QoSQoI.owl#> .
@prefix tl: <http://purl.org/NET/c4dm/timeline.owl#> .

:government a foaf:Organisation, prov: Agent .
:sefki a foaf:Person, prov:Agent ;
      foaf:givenName "Sefki" ;
      foaf:mbox <mailto:s.kolozali@surrey.ac.uk>
      prov:actedonBehalfOf :ccsrSurrey ; .
:sensorRec1 a sao:StreamData, ssn:SensorObservation ;
      prov: wasAttributedTo :government .
:sensorRec2 a sao:StreamData, ssn:SensorObservation ;
      prov: wasAttributedTo :government .
:traffic-sensor-recording-619 a sao:StreamEvent ;
    prov:used [ a sensorRec1; sensorRec2] ;
    sao:time [a tl:Interval;
      tl:at "2014-02-13T08:25:00"^^xsd:dateTime;
      tl:duration "PT15H30M"^^xsd:duration;
    ] ;
    prov:wasAsscoatedWith :sefki ; .
:freshness-traffic-619 a qoi:Freshness ;
      qoi:value "2014-02-13T08:25:00"^^xsd:dateTime .
:sax_AverageSpeedSample a SymbolicAggregateApproximation;
      rdfs:label "The sax representation of the traffic sensor
      recording obtained from Aarhus City.";
      sao:value "bbbbacdd";
      sao:alphabetsize "4"^^xsd:int ;
      sao:segmentsize "8"^^xsd:int ;
      prov:wasGeneratedBy traffic-sensor-recording-619;
      qoi:hasQoI freshness-traffic-619 .
```

Listing 1: A excerpt from an RDF data annotated for a set of sensor recordings given in Figure 7 based on Stream Annotation Ontology.

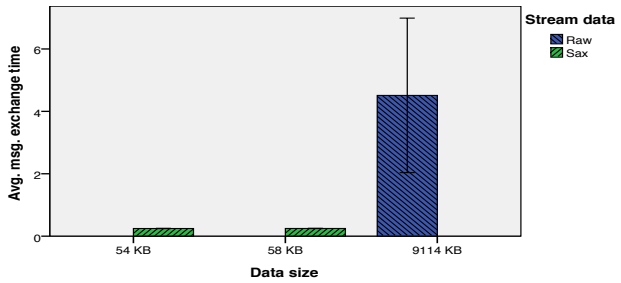
[7]http://dbpedia.org/
[8]http://www.geonames.org/ontology/



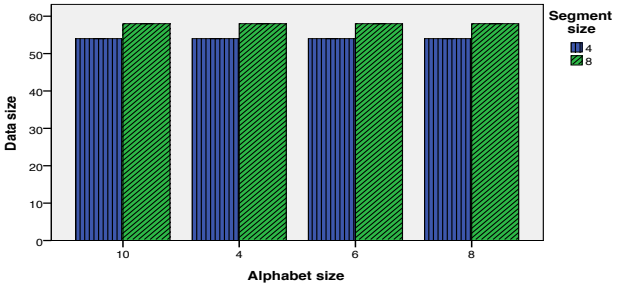
(a) The average message delivery time for each of the number of consumers and data dimensions.



(b) The average message delivery time for each alphabet and segment size of reduced data produced with SAX algorithm.



(c) The average message delivery time and data size for each of the data dimensions.



(d) The data size for each alphabet and segment size of reduced data produced with SAX algorithm.

Figure 8: Summary of the evaluation results for the raw and the sax stream data based on the average message delivery time and data size. The bars refers to the following metrics: number of consumers for each data dimension, namely SAX and raw (Figure 8a); stream data dimension and data size (Figure 8c); the segment and alphabet size for each SAX stream data (Figure 8b and 8d).

MANOVA level of significance are reported using the F-statistics, F, and probability, p.

Performance comparison considering the data size and average message exchange time that were examined using the middleware are reported in Figure 8. For the raw data, the data size and average message exchange time were very high: the data size was 9114 KB and the average time that was spent for message delivery was in range of $2.9s$ to $6.5s$ varying based on the number of consumers. On the other hand, there was a rapid decrease in message delivery time and data size for the sax representation of the data stream. For instance, the data size dropped from its initial high value to 54 and 58 KB, which led to a dramatic decrease to $0.25s$ for the average message delivery time. Intuitively, the difference in message delivery time for various segment and alphabet sizes were very low. For example, the average time difference for the segment size 4 and 8 were $0.01s$, and for the alphabet size it was $0.02s$. Similarly, there was a very low difference (i.e. 4 KB) in data size for the segment and alphabet size. The overall average differences between the raw and summarised data were 96.2% and 99.4% for the data size and average message delivery time, respectively.

The results of the two-way analyses of variance results for the semantic annotation system is reported in Table I. The post-hoc analysis revealed that there was a highly significant

	Data size			Time		
Source	*df*	F	η^2	*df*	F	η^2
DD	1	63808980.48	$1.0^{\star\star\star}$	1	2068152.82	$1.0^{\star\star\star}$
NC	3	0	0	3	69908.90	$1.0^{\star\star\star}$
DD $\times$ NC	3	0	0	3	67934.33	$1.0^{\star\star\star}$

Table I: Results of two-way analyses of variance for the stream annotation system based on the raw and SAX stream data. η^2 is the partial eta squared measure of effect size. $^{\star}p < .05,^{\star\star} p < .01,^{\star\star\star} p < .001$. DD: data dimension; NC: number of consumers.

effect of the SAX representation of the data stream on both the data size and average message delivery time ($p < .001$) with a very large effect size. Therefore, it is evident that dimensionality reduction is not only important for memory space but also for processing time of middleware, which is currently a crucial issue in real-time IoT environments.

On the other hand, there is also highly significant effects of the number of consumers (NC) on the average message delivery time ($p < .001$) with a very large effect size. Even though the overall results for the number of consumers are more or less as reported in [10], the growing delivery time can be explained by the fact that in our experiment the consumers are running on the same machine as the producer which may induce a delay due to low temporary storage

space allocated for each consumer. However, we need to investigate this result with further experiments on a powerful cluster server in future.

The interaction between DD and NC was highly significant for the average message delivery time whereas there was no significant interaction between DD and NC on the data size. Nonetheless, our intuition was solely to examine the effect of interaction between both factors on the middleware, as data reduction is an initial process which cannot be effected by the process of middleware. With these experiments it can be concluded from highly significant interaction of DD and NC that both of the factors have a significant effect on the results, and their effects should be highly considered in real-time and dynamic environments.

VII. Conclusions

In this study, we proposed a stream annotation framework for real time IoT stream using the Advanced Message Queueing Protocol to support delivery of large volumes of data. To represent the summarisation and reliability of stream data, we introduced a new information model that ensures that summarisation techniques can be interpreted as time-based events, even where further semantic associations are unavailable. The framework is tested using different aspects of the stream data, raw and aggregated, in order to find the increase in the performance with our annotated data. The data size and average message exchange time spent through the middleware are used as evaluation metrics. We found that in all cases the framework performance increased 99.4% and 96.2% with annotated summarised stream data in terms of data size and average message delivery time, respectively.

In future work, we will incorporate a wider set of stream data and utilising a computer network, with real-time road traffic to investigate the performance with a large number of consumers in more detail. Further work is also required for the Stream Annotation Ontology to provide a better coverage of stream analysis techniques commonly used by researchers, as well as to enable better generalisation of the model, and harmonisation with existing research tools. This work can help to increase community involvement and will be extended with a vocabulary to cover representation of data analysis features that are created by state-of-the-art stream analysis techniques.

Acknowledgement

The research leading to these results has received funding from the European Commission's Seventh Framework Programme for the CityPulse project under grant agreement no. 609035.

References

[1] Payam Barnaghi, Wei Wang, Lijun Dong, and Chonggang Wang. A linked-data model for semantic sensor streams. *International Conference on Internet of Things (iThings2013)*, pages 468–475, 2013.

[2] Mike Botts, George Percivall, Carl Reed, and John Davidson. Ogc® sensor web enablement: Overview and high level architecture. In *GeoSensor networks*, pages 175–190. Springer, 2008.

[3] Michael Compton, Payam Barnaghi, Luis Bermudez, Raul García-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.

[4] Suparna De, Tarek Elsaleh, Payam Barnaghi, and Stefan Meissner. An internet of things platform for real-world and digital objects. *Scalable Computing: Practice and Experience*, 13(1), 2012.

[5] Frieder Ganz, Payam Barnaghi, and Francois Carrez. Real world internet data. *Sensors Journal, IEEE*, 13(10):3793–3805, 2013.

[6] Cory A. Henson, Holger Neuhaus, Amit P. Sheth, Krishnaprasad Thirunarayan, and Rajkumar Buyya. An ontological representation of time series observations on the semantic sensor web. In *1st International Workshop on the Semantic Sensor Web (SemSensWeb 2009)*, pages 79–94, 2009.

[7] Cory A. Henson, Josh K. Pschorr, Amit P. Sheth, and Krishnaprasad Thirunarayan. Semsos: Semantic sensor observation service. In *International Symposium on Collaborative Technologies and Systems*, pages 44–53. IEEE, 2009.

[8] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1989.

[9] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.*, pages 2–11. ACM Press, 2003.

[10] Sangyoon Oh, Jai-Hoon Kim, and Geoffrey Foz. Real-time performance analysis for publish/subscribe systems. *Future Generation Computer Systems*, 26, March 2010.

[11] Yves Raimond. *A Distributed Music Information System*. PhD thesis, School of Electronic Engineering and Computer Science, Queen Mary University, London UK, 2008.

[12] Steve Vinoski. Advanced message queuing protocol. *IEEE Internet Computing*, 10(6):87–89, 2006.

[13] Wei Wang, Suparna De, Ralf Toenjes, Eike Reetz, and Klaus Moessner. A comprehensive ontology for knowledge representation in the internet of things. In *11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1793–1798. IEEE, 2012.