Semantic Stream Management Framework for Data Consistency in Smart Spaces

Oluwaseun Bamgboye
School of Computing
Edinburgh Napier University
Edinburgh,UK
O.Bamgboye@napier.ac.uk

Xiaodong Liu
School of Computing
Edinburgh Napier University
Edinburgh,UK
X.Liu@napier.ac.uk

Peter Cruickshank
School of Computing
Edinburgh Napier University
Edinburgh,UK
P.Cruickshank@napier.ac.uk

***Abstract*—Semantic technology can provide a bridge between smart applications and Internet of Things (IoT) to enable possible integration and interoperability of data produced by heterogeneous devices. In IoT, data quality plays an important role when it comes to interfacing sensor readings with real-time applications at the basic atomic level. Popular techniques of machine learning and point-based calibrations are inadequate due to inability to perform semantic reasoning and interoperability on sensor streams even in real time. In this paper, a layered software framework based on semantic technologies is developed to maintain the consistency of data streams produced by physical sensors that interprets measurements as numeric values. The framework shows how semantic modelling and reasoning can be applied to validate the consistency of data streams while placing emphasis on the temporal characteristics of the stream. The evaluation of the approach involves analysing the effects of different Resource Description Format(RDF) data serializations on the response times of the reasoning engine and throughput of continuous semantic stream query execution. The outcome of experiments indicates the semantic framework as a promising approach for stream validation in Smart Spaces and other related IoT domains.**

***Index Terms*—Ontology, C-SPARQL, Latency, Throughput, Sensor, Data Stream, Smart Space**

I. INTRODUCTION

Smart Spaces (e.g. smart building) are open pervasive environments having computing hardware with embedded computers and multi-modal sensors providing access to information and unprecedented level of services. They are defined by sensors, measurements, actuators and controllers. The objective of IoT and Smart Spaces concerns integrating heterogeneous readings from different sensors to infer new knowledge for proactive guidance [1].

Sensor readings or measurements are continuous streaming atomic data that are fast changing in space and time. They are used to drive automation, make decisions, execute transactions, entertain, inform and secure. These huge amounts of data produced needs to be consistent and managed quickly for intelligent systems and real-time IoT applications. For instance, sensor measurements derived from temperature and humidity sensors may produce inconsistent readings as a result of physical disturbances or environmental influences. At the same time, these readings may need to be utilised at the lowest atomic level by a highly sensitive context-aware applications such as fire and safety monitoring systems or decision support systems. This will impact on the performance and eventual processing of the system. For this reason, the data needs to be validated to prevent false positive events or malicious decisions from such applications. In this context, it will be necessary to integrate readings from several sensors within the space to perform semantic validation of sensor readings in order to get a coherent idea of the space and to support applications relying on them as infrastructure.

The method of interpolation and integration through semantic modelling and inferencing has been suggested as one of the possible ways of dealing with the data quality problems in sensor readings [2]–[4]. Although semantic technology is becoming popular in stream processing, targeting data quality validation and related issues is marginally unexplored. The previous approaches in dealing with data quality problems in sensor readings mainly adopt either statistical approaches (including analytic techniques) or methods of point calibrations built into sensors. These approaches are not able to provide meaning to sensor streaming data and inadequate to perform real-time semantic reasoning and interoperability on sensor streams.

The contribution of this work is to enhance the stream processing system (such as C-SPARQL) with production rules (using Jena rule constructs) to achieve a Continuous Time-Aware reasoning. This approach is used to provide stream validation for quality problem relating to inconsistencies in sensor streams. It provides semantic meaning and machine interpretation to sensor streams for enhanced sensor streams interoperability.

The rest of this paper is organized as follows: section II discusses the state of the art solutions in semantic technique for sensor streams. Section III presents the semantic software framework for consistency validation of sensor streams. Section IV provides relevant use case scenario that describes the suitability of approach for IoT-based monitoring system in smart building. Section V and VI describes the experiments and Performance Evaluation respectively. The conclusion and future outlook of the work is presented in section VII.

II. RELATED WORK

Sensor-driven IoT applications is a new research area of stream processing system. Until recently the Data Stream Management system (DSMS) has been considered as an area of research for managing and addressing the issues challenging data stream processing. This system is known to be deficient when it comes to performing reasoning over a complex task. In addition, it cannot provide the required protocol to support accessibility and publication of spatial-temporal data [5]. The semantic data modelling and reasoning approach has become an alternative to dealing with sensor streaming data.

The approach to sensor and measurement modelling with the use of ontology model in the domain of Wireless Sensor Networks (WSN) [6] [7] has continue to gain the interest of the semantic web community. The leading standard among these models is the SSN ontology [6], which provides a modular ontology that is considered suitable for various practitioners. It caters for most of the limitation in the previous version which makes it unsuitable for some domain applications. Most ontologies [8] [9] previously developed consider accuracy as being independent of sensor measurements. Although some of these ontologies are able to capture the real time measurements of the sensor readings, the self-evolving capability and important data quality issues relating to Smart Spaces are still lacking and are not within the scope of these models.

The World Wide Web Consortium (W3C) provides a number of reasoning systems to support the process of ontology modelling. These reasoning systems are not able to perform reasoning on the semantic data streams [10]. This shortfall was addressed by StreamRule System [11], which specifically targets the Semantic Web. The system only provides its description using the Extended Mark-up language and fails to support the processing of historical data. SPARQL extensions such as C-SPARQL is currently used as stream querying and processing system for sensor data stream [12] [13]. A new approach to stream reasoning can combines C-SPARQL with either of Descriptive or Non-Descriptive logic systems. Mixed approach requires the combination of Non-Description Logic rule language with stream querying over an ontology [14]. It is necessary to consider a mixed approach that will will implement continuous rule with time-aware feature for data consistency management in Smart Spaces. The current research chooses the option of C-SPARQL against the CQELS because of its support for nested aggregations and temporal operators [15] [16].

Lightweight RDF data serialization is considered the most expressive means for providing meaning and machine interpretations. The most referred RDF data serialization format is the RDF/XML format. This is because it is found to be highly expressive when compared to its counterparts such as N3, Turtle and N-Triples formats. However, recent study [17] indicates RDF/XML serialization can cause increased latency and throughput for centralized reasoning system. Although the claim is yet to be established in the case of sensor streaming data, this forms part of our current investigation.

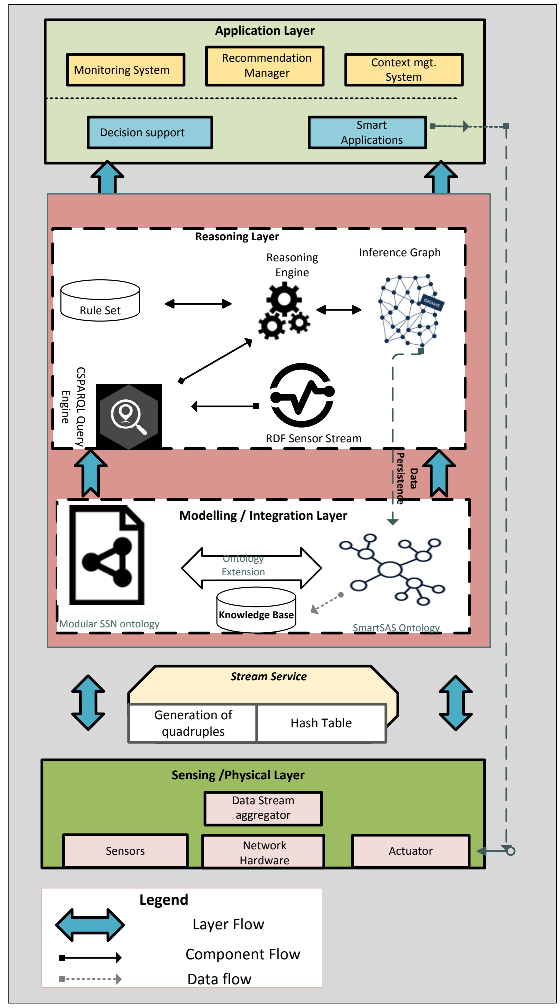

Fig. 1. Semantic framework for stream Consistency.

To the best of our knowledge, combining the Jena Rule construct with C-SPARQL to achieve a continuous rule and time-aware reasoning is yet to be subject to consideration.

III. SEMANTIC FRAMEWORK

In an attempt to provide a method for validating the consistency requirement for sensor streams, we present a semantic framework in figure 1 to describe its implementation as it is related to Smart Spaces. The core of the framework is the semantic process found in the second and third layers . The framework follows a bottom-up approach, where the data entry point is considered to be the first lower layer of the framework. The description of each layer of the framework is similar to the one presented in [4]

A. Sensing Layer

The layer contributes to the large volume of data stream produced within the Smart Space. It consists of sensors and physical network devices. It receives data from the different sensors and devices within the space and prepares the data for the immediate upper layer representing the core of the architecture. Most of the data produced at this layer exists as data streams with temporal characteristics and are heterogeneous nature. In an attempt to arrive at an effective metadata to validate the sensor streams, four different serialization formats for achieving Semantic Stream data are considered [18]. The semantic streams are later constituted as elements of ABox in the modelling layer.

Stream Service

The stream service runs in the background as black box for delivering the specified semantic data serialization formats. It also resolves the continuous binding arising from C-SPARQL query by separating individual quadruples using the hash table indexing key/value property to eliminate redundant quadruples while also providing unique identification for each quadruple. This module is developed with multi-stream management capabilities to ensure the continuous transfer of data streams with java infrastructure such as Apache Camel. It interfaces with a data stream aggregator that collects the raw sensor data and later present them as quadruples (a triple with individual timestamp also called Semantic Stream) to the next upper layer.

B. Modelling and Integration Layer

The method of encoding meaning into raw streaming sensor data useful for other applications is an opportunity for validating the quality of readings. This layer models the ontology for entities of Smart Spaces, description of sensors and streaming data with their relationships. The resulting ontology is called the Smart Space and Sensor Measurement (SmartSASM) Ontology. SmartSASM is used to integrate and enhance reasoning for sensor streaming data available as raw numeric data. The development of SmartSASM extends the latest version of the SSN ontology[1] , which currently considers SOSA [2] (Sensor, Observation, sample and Actuation) ontology at the core to cater for its deficiency. This SSN ontology still lacks self-evolving capability and adequate specification of certain concepts peculiar to Smart Space domain. The logical structure of the SmartSASM ontology reflects extension of fundamental concepts of SSN ontology as identified by the ongoing work in [6]. For example, SmartSASM ontology extends concepts such as 'Feature of Interest' from SSNO with Phenomena ($smartspace : Phenomena \sqsubseteq ssn : FeatureofInterest$), 'Observable Property' with SensorMeasurement ($smartspace : SensorMeasurement \sqsubseteq ssn : ObservableProperty$), and so on.

Consistency Model for Ontology Evolution

We define the consistency model for the evolving ontology based on sensor data stream resulting from frequent update of the persisted data or addition of new concepts. The goal of the evolving property of our ontology is to ensure that the application of changes must result in the ontology conformity to consistency without any loss of data.

[1]http://www.w3.org/ns/ssn/

[2]http://www.w3.org/ns/sosa/

Definition 1. *A single stream STr within a particular time window T_w is consistent iff it maintains specified constraints for each individual time window T_w within each timestamp.*

Since we chose to define the streams consistency for a particular ontology, the set constraints will then depends on the underlying semantic rules. In the remaining part of this section, it is worth noting that most of the definition used for stream consistency model is similar to that in [19]. Then, we can define the stream consistency model M_{STr} as:

$$M_{STr} = C_{Rules} \cup S_{constraints} \cup U_{constraints}$$

Where C_{Rules} are the consistency rules of the model, $S_{constraints}$ are the soft-constraints and the $U_{constraints}$ are the user-defined constraints. These constraints are defined in relation to sensor streams. The same situation holds for the consistency of ontology concepts.

The evolving feature of the ontology is perceived as elementary changes that represents simple, fine-grained changes caused by frequent automatic update performed on the ontology instances. The propagation of this change is closely monitored to maintain and preserve the consistency of the ontology throughout the process. The meta-change transformation of the ontology is best described as *AddConcept* or *AddInstanceOf* depending on the complexity of the change (higher or lower). For example, each concept representing a property or phenomena is automatically populated with its own instance without conflict. Suppose we represent a single Smart space ontology subject to changes as *SpaceOnt*:

Definition 2. *The change Δ between ontologies is a functional mapping of $SpaceOnt_1$ and $SpaceOnt_2$ such that $SpaceOnt_2$ = $\Delta(SpaceOnt_1)$*

It is important to note that in the definition $SpaceOnt_2$ is the changed ontology. This change is tracked along the individual sensor reading. A change is additive when the entities (i.e. concepts or instance) of the resulting ontology are added without altering the existing one. In the present work, instance can added to the existing ontology where new set of sensor reading is produced. Likewise, new concepts can be added automatically where new sensor is detected in the space.

C. Reasoning Layer

The objectives of this layer is to provide semantic reasoning that complement the stream aggregation system (C-SPARQL) in order to achieve a Continuous Time-Aware reasoning. This is used to achieve the process of continuous validation of the sensor stream through the semantic inferencing. Readings from a particular sensor within a time window is validated against prevailing disturbances with data validation policies and, other relevant sensor readings. The fact that most ontology-based reasoners and the counterparts can only provide inferencing for concepts and properties of ontology models [20] is the main rationale for our reasoning system .

1) C-SPARQL Engine:
The query engine is responsible for the continuous aggregation and ordering of semantic streams within each processing window and supports the Stream Service module to perform its intermediate function before the semantic reasoning and inferencing can occur. It adopts the window-based aggregation strategy to perform continuous query over the semantic streams within the sliding windows. The overlapping sensor streams in successive querying window is carefully resolved with customized hash table.

2) Reasoning Engine:
This layer relies on the content of the rule base containing the data consistency policies specified using sets of production rule defined by the domain expert to realize the reasoning task. The rule is applied in continuos fashion to infer new knowledge from the semantic data streams within each sliding window. The Continuous Time-aware feature of the production rule is implemented using Jena rule construct including its Application Programming Interface (API) subsystem because Jena is known be popular for performing read or write operations on application-specific ontology. The rule construct is designed to support the temporal identity of the data streams while putting into consideration the temporal requirements. The execution of Jena rule is influenced by the presence of disturbance(s) within the space. In order to allow the Jena API performs the continuous reasoning over the query window of the quadruples; (i) The conversion of the Jena rule into another system that support reasoning is considered (ii) Layering of Jena rule language with stream processing system (C-SPARQL) to support Continuous Time-Aware reasoning and Close World Assumption (CWA). The problem associated with separation of ontologies during inferencing as identified with Jena is resolved by continuous binding function defined in the framework. The output of the reasoning operation is an inference graph that represents new set of facts. These new facts automatically evolve as updated conceptual model (SmartSASM ontology) in form of historical data.

D. Application Layer

This layer is considered as the uppermost layer of the framework. It consists of various modules that specifies mechanisms for managing interfaces and agents that consumes actionable knowledge produced within the Smart Space. The layer receives output from the reasoning layer as Java Script Object Notation (JSON) format for web based applications. One of the objectives of the layer is to provide support for intelligent services at all levels. The layer can consist of a number of automated applications that interface directly with the underlying semantic process to deliver a better quality output. In order to achieve the desired level of services, the layer relies on the domain logic of the application system.

IV. *Use Case Scenario*

IoT-based Fire and Safety Monitoring System (FSMS) generates alerts to a team of fire service upon detection of fire

```
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
@prefix owl: http://www.w3.org/2002/07/owl#
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema#
@prefix xsd: http://www.w3.org/2001/XMLSchema#
@prefix smartSpace: http://localhost:8080/smartSpace#
[consistencyCheck:
        (?humidityReadings smartSpace:hasHumidityReading ?humidityValue)
        (?humidityReadings smartSpace:humidityHasTimestamp ?humidityTime)
        greaterThan(?humidityValue,39)
        lessThan(?humidityValue,51)
        (?tempReadings smartSpace:tempHasTimestamp ?tempTime)
        (?tempReadings smartSpace:hasValue ?tempValue)
        greaterThan(?tempValue,17)
        lessThan(?tempValue,24)
        (?pressureReadings smartSpace:hasPressureReading ?pressureValue)
        (?pressureReadings smartSpace:pressureHasTimestamp ?pressureTime)
        greaterThan(?pressureValue,750.1)
        lessThan(?pressureValue,761.0)
        le(?tempTime,?humidityTime)
        le(?tempTime,?pressureTime)
        ->
        (?tempReadings smartSpace:isValid 'Consistency Check')
]
```

Fig. 2. Sample Rule Syntax for Sensor Stream Consistency Validation.

from a smart building. The FSMS relies on sensor readings from physical properties within the building to generate notifications, which is sent to a remote command system via HTTP protocol. On event of fire, the firefighters carries a wearable sensor for detecting temperature, smoke, location, etc. One of the fire vehicle contains the on-site command system (CS) - application that continuously evaluates the situation on site to suggest the best strategies for evacuation and rescue operation. The FSMS performs continuous validation and updates on temperature readings in relatively short time intervals by invoking the proposed semantic process against the sensor readings and influencing smart space components within same time window. The remote personnel uses these data to maintain awareness about the situation on site. Similarly the FSMS uses the smart building sensors and other wearable sensors not affected by smoke or fire to deliver streaming data between the site and CS.

V. Experiments and Technologies

The experimental setup consists of a domain/sensor ontology, reasoning engine, continuous query engine and the RDF database. Ideally, The proposed framework from section III has been contextualized in the use case scenario for this experiment. The goal of the experiment is to measure the effects of the semantic data serialization formats on the latency of the reasoning engine and and throughput of the C-SPARQL query.

The dataset from the REFIT[3] Smart Home project has been manipulated for the use case scenario. These data were gathered by 1,567 sensors measuring different properties from 20 homes in city of Loughborough, United Kingdom between 2013 and 2015. It consists of 25,312,397 sensor readings with individual timestamps. The relevant sensor readings from the

[3]http://www.refitsmarthomes.org/index.php/data/

```
REGISTER QUERY sensorValueOf AS
PREFIX smartSpace:<http://localhost:8080/smartSpace#>
SELECT *
FROM STREAM <http://localhost:8080/smartSpace/streamTemperature> [RANGE 15s STEP 2s]
FROM STREAM <http://localhost:8080/smartSpace/streamPressure> [RANGE 15s STEP 2s]
FROM STREAM <http://localhost:8080/smartSpace/streamHumididty> [RANGE 15s STEP 2s]
WHERE {
?tempReadings smartSpace:hasValue ?tempValue.
?tempReadings smartSpace:hasTimestamp ?tempTime.
?tempReadings smartSpace:hasId ?tempId.
?tempReadings smartSpace:hasSeason ?tempSeason.
?tempReadings smartSpace:hasTimestamp ?tempTime.
?pressureReadings smartSpace:hasPressureReading ?pressureValue.
?pressureReadings smartSpace:pressureHasTimestamp ?pressureTime.
?humidityReadings smartSpace:hasHumidityReading ?humidityValue.
?humidityReadings smartSpace:humidityTimestamp ?humidityTime.
}
ORDER BY ASC(?tempTime)
```

Fig. 3. Sample C-SPARQL query.

project were simulated for 24 hours to facilitate the run-time experiments by using the existing Java streaming libraries[4] to generate similar datasets in real time.

In the experiment, we consider four separate semantic serialization formats to represent sensor streams as semantic streams or quadruples within each sliding window. The query engine is allowed to execute concurrently with the streaming data in each window to aggregate multiple quadruples without loss of data points using a sample C-SPARQL queries in figure 3. It adopts a window based aggregation strategy to capture the real time Semantic Streams. The query is able to aggregate an average of 72 quadruples per window.

Furthermore, we induced the system with some likely disturbances and inconsistent temperature readings that can influence the consistent state of temperature readings beyond set point. Such type of disturbances can include *Door/Window Leakage* or *Sensor Status*. This is used as a guide to firing of rule for validating indoor temperature readings and actuation process within the Smart Space. The Jena rule construct (shown in figure 2) is a type of productions that implements a forward RETE system for stream consistency validation of temperature readings. The rule conditions is based on the domain expert knowledge of Occupational Health and safety[5] for ideal indoor temperature readings. The rule was able to validate temperature readings in each sliding window applied to quadruples during reasoning process. The snapshot of the output from the semantic validation is shown in figure 4 where individual data point is represented as quadruple and annotated as *isValid* for true positive temperature readings as seen with *tempReading4*.

VI. Performance Evaluation

The experiment was conducted on a single node centralized server running on multiple processor computer (Pentium Core (TM) i7-4770 CPU @ 3.40GHz – 16GB RAM) with an initial and maximum heap size of 1024m and 2048m respectively. The importance of the semantic querying and reasoning in this work is to detect possible errors or wrong temperature readings from sensor streams and make necessary inferences about the consistent state of the streams in relation to certain external disturbances.

[4]http://streamreasoning.org/resources/c-sparql

[5]http://www.ohsrep.org.au/hazards/workplace-conditions/heat

```
<rdf:Description rdf:about="http://localhost:8080/
        smartSpace#tempReadings4"><smartSpace:isValid
                rdf:datatype="http://www.w3.org/
                2001/XMLSchema#integer">50</smartSpace:isValid>
        <smartSpace:isValid rdf:datatype="http://www.w3.org/
                2001/XMLSchema#integer">54</smartSpace:isValid>
        <smartSpace:hasValue rdf:datatype="http://www.w3.org/
                2001/XMLSchema#float">26.69</smartSpace:hasValue>
        <smartSpace:tempHasTimestamp rdf:datatype="http://
                www.w3.org/2001/XMLSchema#dateTime">2018-09-
                19T20:44:26.160Z</smartSpace:tempHasTimestamp>
        <rdf:type rdf:resource="http://localhost:8080/
                smartSpace#tempValue"/>
</rdf:Description>
<rdf:Description rdf:about="http://localhost:8080/
                smartSpace#tempReadings5">
        <smartSpace:hasValue rdf:datatype="http://www.w3.org/
                2001/XMLSchema#float">11.34</smartSpace:hasValue>
        <smartSpace:tempHasTimestamp rdf:datatype="http://
                www.w3.org/2001/XMLSchema#dateTime">2018-09-
                19T20:44:26.160Z</smartSpace:tempHasTimestamp>
                <rdf:type rdf:resource="http://localhost:8080/
                        smartSpace#tempValue"/>
</rdf:Description>
```

Fig. 4. Sample Inference from Consistency Validation.

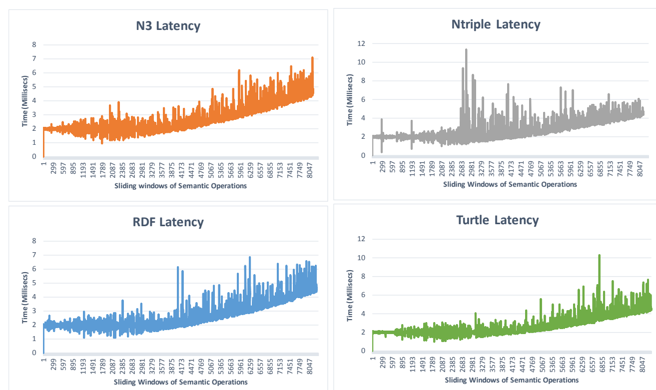

Fig. 5. Latencies of Semantic Process with various Serialization.

In the experiments, we validate the consistency of the temperature readings against the proportionate values and timestamps for humidity and pressure readings while also considering other disturbances and status of sensors within the space.

In the experiment, investigation based on peak times (in milliseconds) of the data serialization formats (i.e. RDF/XML,N-Triple, Turtles,N3) was performed to evaluate the performance of the semantic approach. This was carried out in two specific aspects: (i) the peak mean latency of the reasoning per window to execute the semantic stream by exclusively using each of the four semantic data serialization formats and, (ii) peak throughput for window based aggregation of each of the semantic data formats. The reasoning engine is able to execute more than 8047 window cycles and producing around 656,104 inferred quadruples during the experiments. In specific details, Figure 4 shows the peak of mean latency

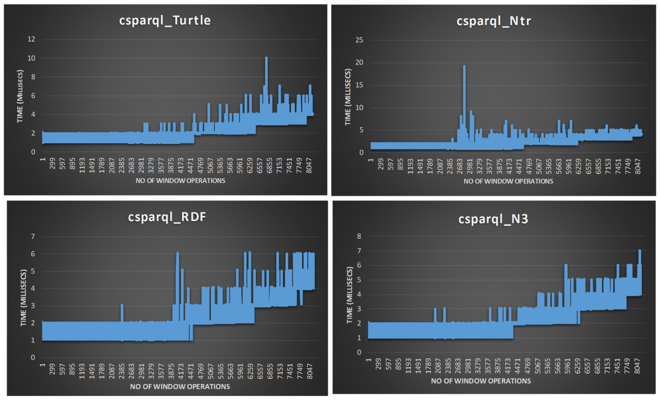

Fig. 6. Throughput of Window-Based Execution of C-SPARQL with various Serialization.

for RDF/XML serialization which is below 7 milliseconds with better performance compared to other formats. This means RDF/XML experiences the shortest delay among other formats. Moreover, Figure 6 indicates both RDF/XML and N3 requires maximum of about 6 milliseconds to perform aggregation over 8163 operations. Considering that streaming data are generated every 15mins (based on the REFIT Smart Home project), it is possible to affirm that the semantic querying and reasoning with RDF/XML serialization can be contained within the interval between read of the participating sensors deployed within Smart Space.

VII. Conclusion And Future Work

Research on stream reasoning that involves the combination of ontologies and query processing has continue to gain the interest of researchers and semantic web community. Most of the researches in the domain of sensors and smart cities deals with various issues but with little concern on the quality of sensor readings. The proposed approach mainly combines domain ontology that provides the description and integration of heterogeneous Smart Space devices/components with semantic reasoning derived from First Order Logic. A remarkable advantage of the approach is that the continuous sliding window aggregation and reasoning technique with Non-Description Logic improves the accuracy of semantic validation over continuous sensor streams. Furthermore, the outcome from the experiments indicates sensor streams can be validated at the basic atomic level of applications before it is consumed by time sensitive or automated applications.

An investigation into the suitability of the framework for validating the sensor readings against plausibility or trustworthiness and incompleteness of sensor streams is being considered. The spatial characteristics of the data stream with the time-awareness is equally worth being considered. In the future, investigation will be conducted in order to confirm the efficiency and effectiveness of the semantic process in distributed environment. Further studies will also include designing the semantic reasoning that is able to handle larger data source and more sophisticated or complex rules.

References

[1] D. O'Sullivan and V. Wade, "A smart space management framework," *Technical Report: Trinity College Dublin*, no. April 2002, pp. 1–23, 2002.

[2] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in internet of things: A state-of-the-art survey," *Journal of Network and Computer Applications*, vol. 73, pp. 57–81, 2016.

[3] D. Kuemper, T. Iggena, R. Toenjes, and E. Pulvermueller, "Valid.IoT: A Framework for Sensor Data Quality Analysis and Interpolation," *Proceedings of the 9th ACM Multimedia Systems Conference*, pp. 294–303, 2018.

[4] O. Bamgboye, X. Liu, and P. Cruickshank, "Towards Modelling and Reasoning About Uncertain Data of Sensor Measurements for Decision Support in Smart Spaces," *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 744–749, 2018.

[5] E. D. Valle, S. Ceri, F. v. Harmelen, and D. Fensel, "It's a Streaming World! Reasoning upon Rapidly Changing Information," *IEEE Intelligent Systems*, vol. 24, no. 6, pp. 83–89, 2009.

[6] A. Haller, K. Janowicz, S. J. D. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation," *Semantic Web*, pp. 1–0, 2018.

[7] K. J. Witt, J. Stanley, D. Smithbauer, D. Mandl, V. Ly, A. Underbrink, and M. Metheny, "Enabling Sensor Webs by Utilizing SWAMO for Autonomous Operations," in *8th NASA Earth Science Technology Conference.*, 2008.

[8] M. Calder, R. A. Morris, and F. Peri, "Machine reasoning about anomalous sensor data," *Ecological Informatics*, vol. 5, no. 1, pp. 9–18, 2010.

[9] M. Compton, "The Semantic Sensor Network: Ontology A Generic Language to Describe Sensor Assets." in *Proceedings of the AGILE 2009 Pre-Conference Workshop Challenges in Geospatial Data Harmonisation, Hannover, Germany*, 2009.

[10] M. A. Aufaure, R. Chiky, O. Curé, H. Khrouf, and G. Kepeklian, "From Business Intelligence to semantic data stream management," *Future Generation Computer Systems*, vol. 63, pp. 100–107, 2016.

[11] A. Mileo and M. Hauswirth, "StreamRule: A Nonmonotonic Stream Reasoning System for the Semantic Web," in *International Conference on Web Reasoning and Rule Systems*, vol. 7994, no. July, 2013, pp. 247–252.

[12] D. Anicic and P. Fodor, "EP-SPARQL: a unified language for event processing and stream reasoning," *Proceedings of the 20th . . .*, pp. 635–644, 2011.

[13] D. F. Barbieri and P. Milano, "Querying RDF Streams with C-SPARQL," *ACM Transactions on Autonomous and Adaptive Systems (TAAS) - Special section on formal methods in pervasive computing, pervasive adaptation, and self-adaptive systems: Models and algorithm*, vol. 7, no. 1, 2014.

[14] E. Garoufallou, R. J. Hartley, and P. Gaitanou, "An Expert System for Water Quality Monitoring Based on Ontology," *Communications in Computer and Information Science*, vol. 544, no. July, 2015.

[15] D. Le-phuoc, "A Native and Adaptive Approach for Linked Stream Data Processing," in *International Semantic Web Conference*, vol. 1380, no. February, 2013, pp. 370–388.

[16] D. F. Barbieri, M. Grossniklaus, and M. Dipartimento, "An Execution Environment for C-SPARQL Queries," in *Proceedings of the 13th International Conference on Extending Database Technology, Lausanne, Switzerland*, 2010, pp. 441–452.

[17] A. I. Maarala, X. Su, and J. Riekki, "Semantic Reasoning for Context-Aware Internet of Things Applications," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 461–473, 2017.

[18] Su, Xiang, Jukka Riekki, Jukka K. Nurminen, Johanna Nieminen and M. Koskimies, "Adding Semantics to Internet of Things," *Concurrency and Computation: Practice and Experience, Wiley*, vol. 18, no. 2, pp. 105–116, 2010.

[19] G. Flouris, D. Plexousakis, and G. Antoniou, "Evolving ontology evolution," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3831 LNCS, pp. 14–29, 2006.

[20] G. D'Aniello, M. Gaeta, F. Orciuoli, and A. Simonetti, "An approach based on semantic stream reasoning to support decision processes in smart cities," *Telematics and Informatics*, vol. 35, no. 6, p. 1795, 2018.