Future Network & MobileSummit 2013 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2013
ISBN: 978-1-905824-36-6

Supporting Smart-city Mobility with Cognitive Internet of Things

Andrey SOMOV, Corentin DUPONT, and Raffaele GIAFFREDA
CREATE-NET, Via alla Cascata 56/D Povo, Trento (TN), 38123, Italy
Tel: +39 0461 408 ext. 406, Fax: + 39 0461 421157,
Email: {asomov; cdupont; rgiaffreda}@create-net.org

Abstract: A Wireless Sensor Network (WSN) is a key data provider for the Internet of Things (IoT). A WSN can serve as a tool for both identification and data generation. However, due to its inherent resource limitation the WSNs cannot generate and transmit large data streams and, therefore, typically transmit raw and simple sensor values. Furthermore, the sensors usually transmit their data in proprietary formats to an embedded application. This can be enough for WSN control and monitoring applications, but is not enough for the IoT where it is expected that thousands of different objects belonging to different context will be accessed remotely. In this work we propose to create a virtual representation of real objects (sensors) with a corresponding Virtual Object (VO) model. This VO produces not solely a stream of raw sensor measurements, but enriches those with context information. We evaluate our approach using a real city-scale traffic monitoring sensor network deployed in the city of Enschede, the Netherlands.

Keywords: Internet of Things (IoT), Resource Description Framework (RDF), traffic monitoring, smart city.

1. Introduction

In the last decade, the Internet of Things (IoT) paradigm [11] has slowly but continuously conquered the minds of researchers and engineers expected, to the point of becoming one the most exciting innovation domains as shown by the hype at recent International Consumer Electronics Show[1] (CES). Underneath such hype there are a number of enabling 'pillar' technologies: sensor and actuator networks, identification and tracking technologies, enhanced communication protocols, distributed intelligence, and cognitive technologies. Amongst these, Wireless Sensor Networks (WSN) [12] are considered to be the key data provider for the IoT. A WSN is a collection of tiny, autonomous sensor nodes which measure physical conditions [14] and send the results over the wireless network to a WSN gateway.

The last two decades were characterized by a tremendous progress in WSN design fostering their increasingly widespread usage and leading to growth in both, number of nodes per deployment and infrastructure complexity. Nowadays, a city-scale WSN deployment[2] is not a novelty anymore. The city-scale deployments can include a number of sub-deployments or be a single one. These deployments generate lots of data of different type which has to be synchronized, interpreted, adapted for specific needs, interconnected with other data and/or distinguished in service and application data. However, the data generated by typical WSNs are scanty in terms of 'richness': due to its inherent resource limitation the wireless sensor nodes cannot generate and transmit large data streams. With respect to this a number of open problems can be identified:

[1] International consumer electronics show, www.cesweb.org
[2] Smart Santander project, www.smartsantander.eu

- Heterogeneity: data received from different sensor nodes may be in different formats since the devices may use different software and hardware.
- Data poorness: enrichment of the raw WSN data with metadata and data proper management, will result in a more efficient data exploitation, e.g. data re-use, standardized data storage.
- Data unavailability: WSNs are typically locked into unimodal closed systems, therefore, limiting the access to sensed data.

Heterogeneity, raw data and unavailability are the problems we attempt to address with this work, by proposing a framework for processing, enriching raw data and storing it according to a given information model fostering standardised interfacing to sensed data and interoperability with various applications.

In particular, we use crossroad traffic monitoring data from the WSN and induction loops installed on crossroads in the city of Enschede, in the Netherlands. To achieve the results presented in this paper we created a virtual representation of each sensor with a corresponding virtual object model. Compared to the real sensor, the virtual object produces not only a stream of sensed raw values, but it enriches those with context information, e.g. time and date, location, type of sensor. This information extension allows us to broaden the potential usage of the data in a variety of different applications empowered by cognitive technologies that exploit such enrichment.

This paper is organized as follows: we first introduce the reader to the state-of-the-art works in the field in Section 2. In section 3 we briefly describe a real sensor network deployment in the city of Enschede. We then present the concept of virtual object and its information model in Section 4. The procedure of raw data transformation into RDF documents in described in Section 5. We demonstrate how to enrich the data and extract "knowledge" from data in Sections 6. Business benefits and summary of our work is provided in Sections 7 and 8 respectively.

2. State-of-the-Art

The number of devices which generate data, e.g. sensors, cameras, constantly grows. The network administrators can not manually to manage large data streams from heterogeneous sources. Under the 'management' we understand the data processing procedure which stores the outcome of processing in Machine-to-Machine (M2M) understandable format. To address the issue of automatic data management a number of approaches have been proposed recently. In fact, these approaches can be divided into three groups:

1. Data management onboard, when a sensing device processes and manages raw data stream on board and forwards the ready-to-use data in M2M format to the user,
2. Local data management, when raw data stream is received by a base PC/server, processed and registered,
3. Remote data management, when data stream is received by a web service and remotely processed and managed; a user in this case can retrieve raw data as well as processed data.

Data management onboard. Storing data onboard of sensor nodes is a complicated task especially in the context of the IoT with its heterogeneous sensing platforms, their programming style, and resource constraints. Here we briefly review two techniques on embedded devices self-description.

Web Services Description Language (WSDL) is used in [3] to enable self-description of embedded web services. These documents are properly compressed to meet strict requirement on long-term operation of sensor nodes and stored on board. Given work supports WSDL documents retrieval and discovery by using embedded discovery protocol. The description of sensor nodes using WSDL does not provide an opportunity for standardized integration of self-described sensor nodes with the linked data cloud.

This problem is addressed in [1] where the authors propose an approach to create RDF documents right on embedded IoT devices such as sensor nodes. To overcome sensing devices heterogeneity, the authors use Wiselib [2] which is the library of algorithms for platform-independent development on embedded systems, e.g. sensor nodes. The bottom line of this approach is that devices possess the self-description capability which allows them to describe, for example, the available sensors and services in RDF format. In this case these devices operate as the automatic semantic data generators. Using this approach one can link various RDF data forming data clusters or large data sets. In spite the authors use an RDF file compression technique, this approach is considered to be a power 'hungry' since a transmitter and receiver will be used for a longer period to transmit the RDF file rather than just sensed values.

Local data management. In contrast to 'onboard' approach, in the 'local' one the data and their semantic description are stored on a base computer or server.

For example, [4] presents an infrastructure for data processing in large-scale interconnected sensor networks which is also referred as Global Sensor Network (GSN). The key notion in GSN is the virtual sensor which can be any data generator including a video camera, real sensor, cellular phone or a combination of virtual sensors. The virtual sensors encode their data stream in XML format. To enable easy deployment and usage of virtual sensor, its specification provides all necessary information: metadata used, structure of the data stream, SQL-based specification of the stream processing in a virtual sensor, and functional properties. Using this approach one can 'virtualize' large networks.

Similar approach proposed in [5] where single data generators are referred as Virtual Objects (VO) and add to the proposed framework cognitive mechanisms. The authors semantically enrich virtual counterparts of ICT Real-World Objects and propose to associate non-ICT RWOs, e.g. furniture, room, person, using ICT ones. Besides that, in accordance with application request single VOs can be organized in Composite Virtual Objects (CVO) with cognitive functionalities, e.g. situation acquisition, self-x functions, reasoning.

Smart identification framework for ubiquitous computing is presented in [6]. The objects are identified with RFID devices as in CASAGRAS [7] and are described with minimal metadata. As soon as virtual counterpart is created the application communicates with it not trying to access real object. In the case when real objects does not exist any more the framework destroys its virtual copy. The work in [6] follows the object representation strategy whereas CASAGRAS relies on ontology.

Remote data management. Web services such as Cosm (former Pachube) [8] or Sen.Se [9] create virtual objects for the IoT. These services accept raw data from sensors or other services and post them. Each feed has limited description in terms of metadata, e.g. location, measurement units, data owner, and limited number of samples to be stored, i.e. new data which exceed the storage threshold automatically remove the first posted data. This approach helps to open the data to community and simplifies the access to them: one need have the credentials and a simple Python/Java script. For example, research work in [10] uses Cosm for posting sensed values as well as posting actuation commands based on processed sensed data.

3. Deployment

In this section we briefly discuss traffic monitoring scenario and present the network architecture for traffic data collection in the city of Enschede.

3.1 – Scenario

Figure 1 shows the map of sensors deployed in the city of Enschede. For example, 50 out of 75 intersections marked with red circle (here we mean the intersections with traffic lights) are equipped with inductive loops. These inductive loop act as sensors and count vehicles passed by. To ensure accurate vehicle detection the lanes typically have three detectors per signal group. For example, if a vehicle is not detected by a loop or due to some reasons it does not pass the loop, there are two other loops in the lane to detect the vehicle. Also, a vehicle can be detected by the loops of adjacent signal groups in the case when it changes the lane. The controller (one per intersection) collects the data from detectors in order to (i) locally manage the traffic lights based on the data collected from the inductive loops and (ii) forward the collected data to a remote server for further processing.

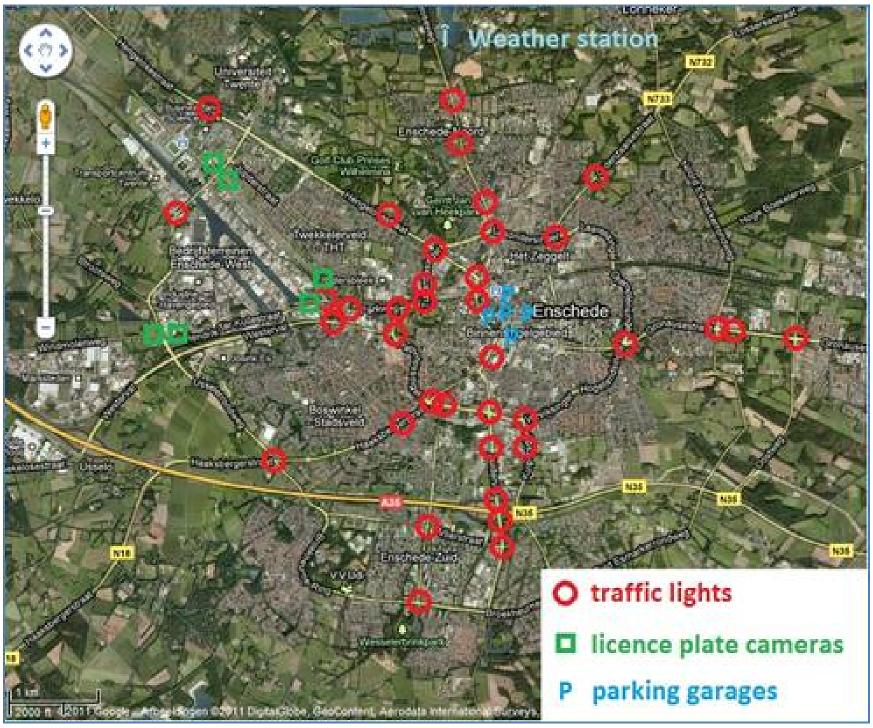

Figure 1. The map of WSN deployment in the city of Enschede, the Netherlands.

In this work we use only the data received from the traffic lights. Next section describes in more details the hardware architecture of sensor network for raw data collection.

3.2 Hardware architecture

Network architecture for traffic data collection in the city of Enschede, the Netherlands is comrised of five layers: sensors (inductive loop detectors), controllers, central hub, central server, and user layer.

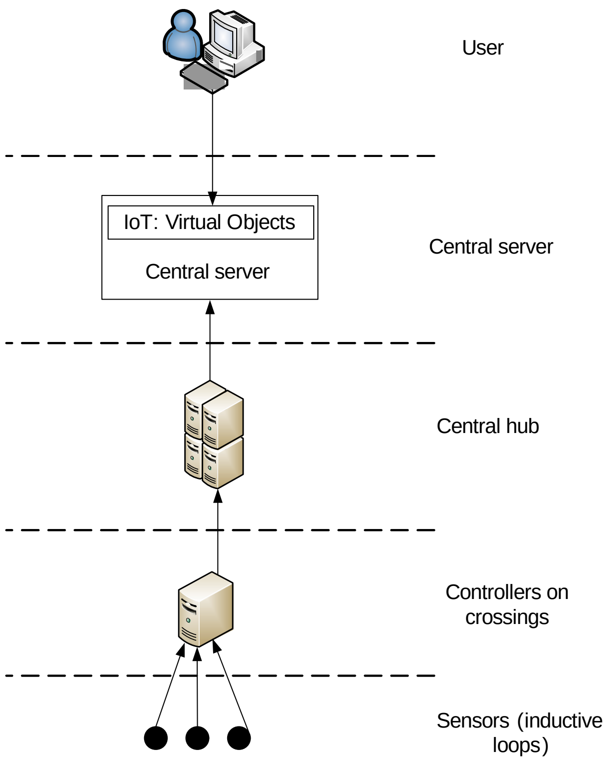

Figure 2. Schematic architecture of network for traffic data collection in the city of Enschede, the Netherlands.

Sensors. An inductive loop is an electromagnetic detection system which uses a moving magnet to induce an electrical current in a nearby wire. It enables the detection of a vehicle passing or arriving at a certain point. This data are forwarded to controllers located at the intersections near the traffic lights.

Controllers. Each signalized intersection has a controller which collects data from detectors. The main goal of controllers is to capture the movement of a vehicle. Controllers, however, are not smart devices and can not predict route patterns, perform inference procedures or tracing back. Upon detecting the vehicle their tasks are pretty straightforward: (i) forward measured values over the network and (ii) locally control the respective traffic lights based on the collected data and/or standardized phase sequences.

Central hub. The role of central hub consists in collecting the traffic data from the controllers and its transfer to the central server.

Central server. The server is served by a local networking company which takes care about security and privacy issues. This server stores the data from all controllers. The data received from the controllers contain raw data and metadata. We discuss data in more details in next section.

Virtual Objects (VO) created in the context of the IoT are also stored on the central server. Users can access the VOs for getting the up to date traffic information and/or use the VOs to create applications and services. The following sections describe how we create VOs and transform received raw data in RDF documents.

4. Virtual Object and Information Model

A VO is the virtual (abstract) representation of an ICT object that may be associated with a non-ICT object. Indeed, the act of 'installation' brings the ICT object in a specific real-world context. VOs indeed help in accessing the real world objects and helps interfacing them (after abstraction) to the external world. The features, functionalities and resources (e.g. memory, computation, communication etc.) represented by a VO can be accessed and re-used by other entities in the Enschede traffic system. Once a VO is installed, it may also be associated with one or more non-ICT objects. Information about the status of non-ICT objects is gathered through sensing capabilities of the ICT object, while the status of objects can be manipulated using actuation capabilities of ICT objects.

The VO vision allows a vast amount of heterogeneous devices to be part of a versatile IoT service platform, while providing contextualised, meaningful information on real-time status of Real World Objects (RWOs). Whenever the status or situational context of a RWO changes, this is reflected in the VO. The term Real World Object (RWO) refers to any object that exists in the real/physical world. The RWOs might be classified as ICT Objects/devices (e.g.: a device such as a sensor, actuator, Smartphone, etc) and non-ICT Objects (e.g.: a room, a person, a car, a tire of a car, a strawberry, etc.). An ICT object may be located at a certain physical location, may be associated to a non-ICT object and may offer one or more functions (e.g. temperature measurements, humidity measurements, luminosity measurements, location of an object/person, etc). Non-ICT objects correspond to objects or entities of the physical world that do not have any direct ICT capabilities such as furniture, a room, fruits, a person, a city, etc. Further, the digital representation of the RWO is called a Digital World Object (DWO).

The concept of VO is of vital importance in the context of smart city and the IoT [13]. It helps one to overcome the problems of devices heterogeneity, scalability and enrich the raw data generated by the devices with metadata (context information). Moreover, using machine learning techniques a user can “substitute” real sensors with virtual ones. Finally, VOs can be accessed anytime from anywhere.

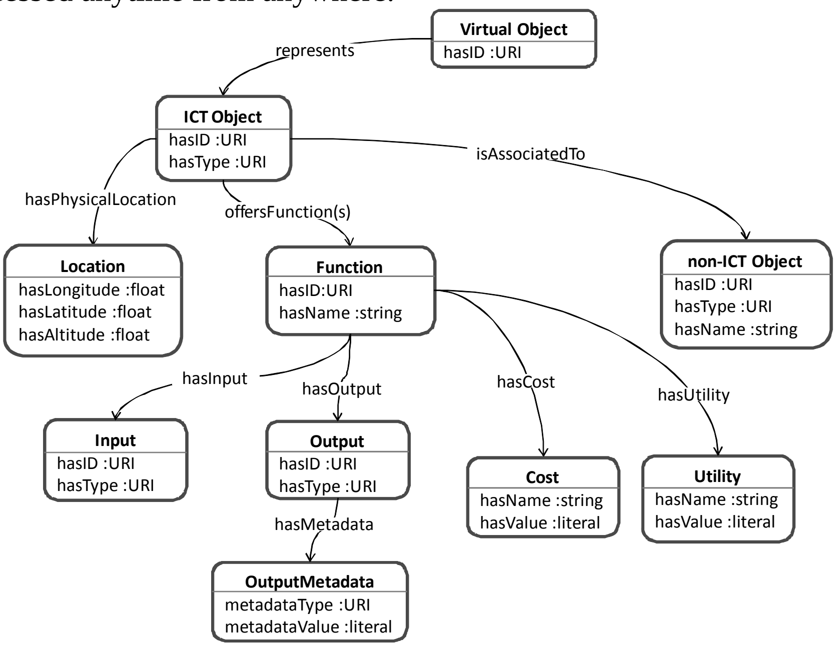

Figure 3. VO information model.

An ontology has been developed in order to be used as information model (see Figure 3) for the description of the VOs. The Resource Description Framework (RDF) has been used in order that the information to be presented as machine readable/understandable data. The mechanisms retrieve the available parameters form the VO Templates and store them as RDF Triples in form Subject – Predicate – Object. The semantically enriched information, are stored in the VO registries that have been implemented as RDF graph databases with the use of Sesame, an extensible Java framework that supports the management of the RDF data. The SPARQL query language used to allow the interaction with the information that are stored in VO registries. Creation of RDF triples and their retrieving using SPARQL is presented in next section in more details.

5. Data Transformation

This section describes how raw sensor data can be enriched and transformed into Resource Description Framework (RDF) format compliant with information model presented in previous section. RDF document can be then published in a database as well as requested by an application. A reader can also refer to the alternative data transformation solution [15]. In our vision, however, it lacks the implementation simplicity.

First, we would like to clarify why the ultimate goal of such a conversion is the RDF format. The RDF data model has become the actual standard for the description of real-world phenomena and has no dependence on an application. The main idea of RDF consists in making statements about resources in the form of subject-predicate-object expressions. This allows linking described resources with other RDF data.

The data transformation consists of two stages (see Figure 4). The goal of the first stage is to get XML file, the goal of the second – RDF file. We can receive raw data from the Central Server (see Section 3.2) in Enschede in CSV or HTML format. In this work we consider two scenarios of raw data:

- 'Travel time per day': license plate cameras installed along three main directions in Enschede count the number of vehicles;
- 'Intensity': Induction loops measuring passing vehicles near the traffic lights.

The raw data are distributed in the columns and show the number of cars detected per 15 minutes within 24 hours. File name is comprised of data and route the measurements are collected. In order to get XML file we create XML schema (XSD file) and XLS (Excel) document. XML schema shows which data (and their types) will be included in RDF and they are linked and interconnected. We note here that this document does not contain any raw data, i.e. values received from the sensors. At this stage a user may enrich the XML schema with any types of data he needs to have in the final document. XLS document, in contrast, must contain raw data distributed in columns, for instance, where columns must have the headers' names with respect to data types defined in XML schema. Upon completion of both documents XLS to XML mapping exists using XML schema.

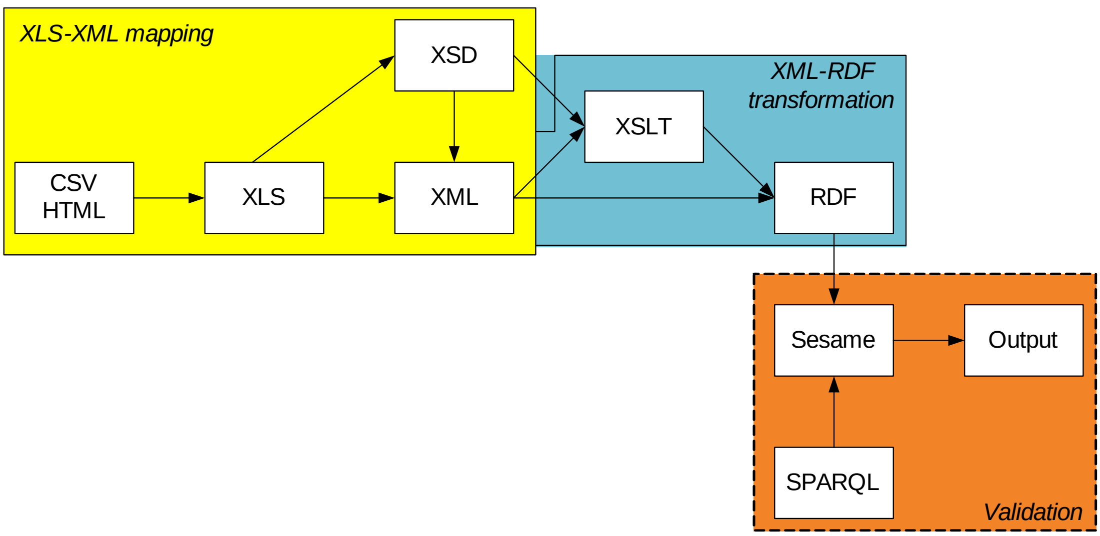

Figure 4. Sensor data transformation to RDF, its publishing and requesting.

To complete the transformation XML and XSLT files are required. Extensible Stylesheet Language Transformations (XSLT) is a language which helps to transform an XML file into other XML-based documents. To do so we used 'xml2rdf3.xsl' file available on Github. For the XML schema creation and XML-RDF conversion we used XMLSpy sofware by Altova. RDF file has to be uploaded to Sesame data base.

6. Knowledge Creation and Data Enrichment

In this section we demonstrate a technique able to convert “Data” into “Knowledge”. The Data can be seen as raw information extracted from sensors, whereas the Knowledge is information that is consolidated, contextualised, and less transient than Data. For example, the information that a street is currently blocked by a traffic jam is considered as knowledge. This information is derived from the analysis on the sensor data located in and around this street. By essence, Knowledge is more reusable than raw Data because it is more contextualized, and more stable in time. It also obviously presents more value for a user.

As a result of the process presented in the previous chapter, an RDF file as been created containing the raw data from the sensors, and this data has been uploaded in a Sesame data base. As an example, we’ll use the data provided by traffic light sensors in the city of Enschede, able to count the numbers of cars passing. Those traffic light sensors are located in 4 different axis (Noord-Zuiderval, Zuiderval-Singels, Gronausestraat-Euregioweg, Gronausestraat-Boulevard) for a total of 40 traffic lights and corresponding sensors. The data provided corresponds to a car count every 15 minutes.

As a first step, we have to enrich the data provided with context information, such as the geographical position of the traffic lights. This is done with the following SPARQL update:

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
INSERT DATA {
<ns1:EIT_Experiments/ns1:IntensityFromTrafficLights/ns1:one>
   geo:lat "55.701";
   geo:lon "12.552".
}
```

Figure 5. Enriching data with GPS coordinates for the traffic lights.

This operation attributes a latitude and longitude to the first traffic light named “one” and insert them in the RDF data base. It must be done for every traffic light.
We can now query our database for the number of vehicles that passed by a certain traffic light from 18:00 to 20:00:

```
SELECT (SUM(xsd:decimal(?carCountValue)) as ?total)
WHERE {
   ?IntensityFromTrafficLights ns1:TimeFrom ?timeFrom.
   ?timeFrom rdf:value ?time.
   ?IntensityFromTrafficLights ns1:one ?carCount.
   ?carCount rdf:value ?carCountValue.
FILTER (?time >= "18:00:00.000" && ?time < "20:00:00.000")
}
```

```
Result: 293
```

Figure 6. Consolidating the number of cars.

We then present a technique of Knowledge creation and enrichment based on SPARQL using the keyword CONSTRUCT. Indeed, whereas a SELECT query on an RDF graph returns the results formatted as a table, a CONSTRUCT query is returning the results formatted as an RDF graph themselves. For example one can insert a rule on traffic jam detection directly in the RDF graph as a CONSTRUCT request. Then when a user will query for traffic jams, this rule will be triggered. The request is the following:

```
CONSTRUCT {?carCount ns1:HasTrafficJam "Jammed".}
WHERE {
   ?IntensityFromTrafficLights ns1:TimeFrom ?timeFrom.
   ?timeFrom rdf:value ?time.
   ?IntensityFromTrafficLights ns1:one ?carCount.
   ?carCount rdf:value ?carCountValue.
FILTER (xsd:decimal(?carCountValue)>100)
}
```

Figure 7. Creation knowledge on traffic jams.

In this request we construct a new RDF graph containing the “jammed” status of every traffic light. We arbitrarily consider that a traffic jam has appeared if more than 100 cars have passed at a traffic light in the last 15 minutes. This new RDF graph comes as a complement from the original RDF graph containing the initial data and can be, in turn, queried.

Furthermore, this SPARQL request can be stored itself as RDF data, using the SPIN SPARQL dialect[3]. SPIN SPARQL defines an RDF representation of SPARQL. The original RDF data is thus enriched with “rules” stored themselves as RDF. This has the advantage to save a lot of space in database: a rule is often more concise than the data it generates. It also allows to store all resources from a domain model (both rules and data) under the same representation and in the same place, which enhance reusability.

7. Business Benefits

The presented work and results are expected to bring considerable value to future smart-cities and Intelligent Transport Systems (ITS) applications. Ordinary cities are already today characterized by a wealth of data being collected through a number of different sensing infrastructures and partly used for different, often siloed application purposes.

Putting aside bespoke applications and associated sensing infrastructures that particular cities may have, with the Internet of Things becoming more and more established, as anticipated in the introduction, the trend we can currently observe is an increasing amount data enriching the overall set and coming from connected objects and sensor.

With initiatives dedicated to opening-up such data aiming to promote innovative use and useful applications creation, our work aims at lowering the entry-level threshold for developers, as it gives the means to semantically enrich and also aggregate data in ways that the application stakeholders can then exploit.

Understanding how and with what extra info to enrich sensed data and publish it as more composite RDF data can also be of interest for sensor manufacturers, which can in this way gain differential advantage for tapping into the business potentials of cognitive IoT.

8. Conclusions and Future Work

In this work we have proposed to enrich raw sensor data with context information for further usage in the IoT applications. To do so, we have developed virtual representations of real sensors called virtual objects. These IoT objects produce not only raw sensor measurements, but enrich those with context information as well. The enriched data are stored in RDF documents. Our experimental scenario where we evaluated our approach included real sensor network deployment for traffic monitoring in the city of Enschede. In particular, we have demonstrated how to enrich raw data and how to extract ‘knowledge’ from the available data. The experimental results demonstrate high potential of IoT paradigm towards successful traffic monitoring in the context of smart city.

[3] http://spinrdf.org/sp.html

Our future work includes full automatization of VO creation for traffic monitoring use case. However, a reader can refer to our recent work [13] where we evaluate the time of VO creation using simulations in the context of proposed IoT framework. Besides, we plan to employ machine learning techniques to performs inference procedures, e.g. create route patterns, predict road congestions.

Acknowledgements

The authors would like to thank Erik Klok, Marcel Meeuwissen (Municipality of Enschede, the Netherlands), and Sander Veenstra (University of Twente, the Netherlands) for providing real traffic data and fruitful discussions.

This paper describes work undertaken in the context of the iCore project, 'Internet Connected Objects for Reconfigurable Ecosystems' (http://www.iot-icore.eu/). iCore is an EU Integrated Project funded within the European 7th Framework Programme, contract number: 287708. The contents of this publication are the sole responsibility of iCore project and can in no way be taken to reflect the views of the European Union.

References

[1] H. Hasemann, A. Kroller, and M. Pagel, "RDF Provisioning for the Internet of Things," In *Proceedings of the Internet of Things 2012 International Conference (IoT 2012)*, pp. 143-150, Wuxi, China, October 24-26, 2012.

[2] T. Baumgartner, I. Chatzigiannakis, S. P. Fekete, C. Koninis, A. Kroller, and A. Pyrgelis, "Wiselib: A Generic Algorithm Library for Heterogeneous Sensor networks," In *Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN 2010)*, pp. 162-177, Coimbra, Portugal, 17-19 February, 2010.

[3] N. Glombitza, R. Mietz, K. Romer, S. Fischer, and D. Pfisterer, "Self-Description and Protocol Conversion for a Web of Things," In *Proceedings of International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp.229-236, Newport Beach, California, USA, 7-9 June, 2010.

[4] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks", Proc. of *the International Conference on Mobile Data Management (MDM 07)*. Washington, DC, USA, May 2007, pp. 198-205, doi: 10.1109/MDM.2007.36.

[5] D. Kelaidonis, A. Somov, V. Foteinos, G. Poulios, V. Stavroulaki, P. Vlacheas, P. Demestichas, A. Baranov, A. R. Biswas, R. Giaffreda, "Virtualization and cognitive management of real world objects in the internet of things," In *Proceedings of the International Conference on Internet of Things*, pp. 187-194, Besançon, France, November 20-23, 2012.

[6] K. Romer, T. Schoch, F. Mattern, T. Dubendorfer, "Smart Identification Frameworks for Ubiquitous Computing Applications," J. Wireless Networks 10(6): 689-700, 2004.

[7] CASAGRAS2, http://www.iot-casagras.org

[8] Cosm platform, https://cosm.com

[9] Sen.Se platform, http://open.sen.se

[10] D. Kelaidonis, A. Somov, G. Poulios, V. Foteinos, V. Stavroulaki, P. Vlacheas, P. Demestichas "A cognitive management framework for smart objects and applications in the internet of things," In *Proceedings of the 2nd Workshop on Smart Objects Resource Management*, Hamburg, Germany, September 26, 2012.

[11] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtach, "Internet of Things: Vision, Applications and Research Challenges," Ad Hoc Networks, 10(7): 1497-1516, 2012.

[12] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," IEEE Comput. 37(8): 50–56, 2004.

[13] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, A. Somov, V. Foteinos, G. Poulios, A. R. Biswas, K. Moessner, P. Demestichas, "Enabling smart cities through a cognitive management framework for the internet of things," IEEE Communications Magazine, to appear, 2013.

[14] A. Somov, A. Baranov, A. Savkin, D. Spirjakin, A. Spirjakin, and R. Passerone, "Development of wireless sensor network for combustible gas monitoring," J. Sensors and Actuators, A: Physical 171(2): 398-405, 2011.

[15] Model driven engineering, http://wiki.eclipse.org/ATL/Concepts