Semantic Multi-sensor Data Processing for Smart Environments

Fano Ramparany
Orange Labs, 28 chemin du Vieux Chene, 38243 Meylan, France

Keywords: Semantic Web, Internet of Things, Smart Home, Smart Sensors.

Abstract: One salient feature of data produced by the IoT is its heterogeneity. Despite this heterogeneity, future IoT applications including Smart Home, Smart City, Smart Energy services, will require that all data be easily compared, correlated and merged, and that interpretation of this resulting aggregate into higher level context better matches people needs and requirements. In this paper we propose a framework based on semantic technologies for aggregating IoT data. Our approach has been assessed in the domain of the Smart Home with real data provided by Orange Homelive solution. We show that our approach enables simple reasoning mechanisms to be conducted on the aggregated data, so that contexts such as the presence, activities of people as well as abnormal situations requiring corrective actions, be inferred.

1 INTRODUCTION

An IDC study (idc, 2015) predicts that the number of connected objects is approaching 200 billion today with 7% (14 billion) already connected to the internet. Most of these objects automatically record, report and receive data. Although the volume of these IoT data currently represents only 2% of the world's data, the same study report that by 2020 it will increase up to 10%.

This data has been characterized by IBM data scientists along four dimensions: volume, variety, velocity and veracity.

In this paper we mainly address the Variety issue, which further refer to incompatible data formats, non-aligned data structures and inconsistent data semantics.

IoT data is heterogeneous both semantically (the temperature in my bedroom doesn't have much to do with the positioning of my fridge in the kitchen) and syntactically (a temperature is a floating point number expressed in celsius degrees, whereas a position is a coordinates pair expressed in meters with respect to some defined reference origin). Despite this heteogenity, future IoT applications including Smart Home, Smart City, Smart Energy services, will require that all data be easily compared, correlated and merged and that interpretation of the resulting aggregate into higher level context better matches people needs and requirements, bringing user experience at the next level. In this paper we propose a framework based on semantic technologies to aggregate IoT data. Our approach has been assessed in the domain of the Smart Home with real data provided by Orange Homelive solution. We show that our approach enables simple reasoning mechanisms to be conducted on the aggregated data, so that contexts such as the presence, activities of people as well as abnormal situations requiring corrective actions, can be recognized and properly handled.

In the next section we state the problems and draw the related state of the art. We then introduce the experimental platform that we used, to develop and assess our solution approach. This will enable us to illustrate the technical and scientific challenges that we face with a real Smart Home setting. We then develop our semantic modeling approach and elaborate on the benefit of this approach in terms of reasoning and high level interpration that this model allows. We finally discuss our approach with its short terms perspectives and will unveil a first repertoire of use-cases exploiting our approach that will improve the experience of Smart Home occupants.

2 PROBLEM STATEMENT

In a previous study we already pointed out this issue and named it "aggregation of heterogeneous pieces of information" (Ramparany et al., 2014). We mentioned this recommendation use case: Suppose that we could gather in the same model, the weather fore-

Ramparany, F.
Semantic Multi-sensor Data Processing for Smart Environments.
DOI: 10.5220/0005729601990206
In *Proceedings of the 5th International Confererence on Sensor Networks (SENSORNETS 2016)*, pages 199-206
ISBN: 978-989-758-169-4

cast for the next 4 hours, a person geographical location, her/his preferences/profile/activities and the list of public swimming pools in the area. Having these multiple information in a single place makes it easy to reason upon and for instance to produce personnalized recommendation such as enjoying a dip in the nearest swimming pool if the person is available and enclined to do so.

Some work has been conducted in analyzing the benefit of semantic modeling in the domain of pervasive computing ((Ramparany et al., 2007), (Sorici et al., 2015), (Ye et al., 2015)), but to our knowledge none have pushed to the point of implementing and evaluating it on real life data.

The value of semantic technologies has been recognized for sometimes now for integrating database schema, data modeling and processing.

2.1 Semantic Data Integration

Data integration research has been focused in database schema integration approaches and the use of ontologies and related semantic technologies to provide data consistency among heterogeneous database schemas. The theoretical foundations of this Ontology-Based Data Access (OBDA) (Lenzerini, 2011) have been thoroughly investigated. Prototypical implementations have been also conducted such as Quest (Rodriguez-Muro et al., 2012) or MASTRO (Calvanese et al., 2011). As a matter of fact, internally, ontologies will be based on DL-Lite logic which essentially captures standard conceptual modelling formalisms, such as UML Class Diagrams and Entity-Relationship Schemas, and are at the basis of OWL 2, the current W3C standard language for ontologies (W3C,).

The Web since its origins has been a vehicle of data interchange. However, automatic discovery and integration of Web data has been impractical until the availability of the RDF framework and RDF data sources. The flagship initiative on this area, Linked-Data (Berners-Lee, 2006) has fostered both the size of the structured Web data and its exploitation (Bizer et al., 2009). One of the pillars of this idea is the possibility of retrieving specific data in the web of data; this task is performed by SPARQL (Hartig et al., 2009), a SQL-like language that enables querying a RDF store. Also, the Web currently explores other approaches based on embedded JSON information or microformats, using the tag facilities for HTML. In particular, a specific syntax for using JSON called JSON-LD has ben recently introduced to serialize LinkedData with the motivation to reduce the size of RDF documents compared to the size yielded by XML serialization.

2.2 Semantic Data Modeling

One major benefit of expressing data representation with semantic langage relates to its ability to provide high level and expressive abstractions. For instance, in the IoT, data abstraction is concerned with the ways that the physical world is perceived and managed. In this domain, a Semantic Sensor Network ontology (Compton et al., 2012) has been developped and proposed at the W3C for standardization.

This vision of introducing abstraction based on a semantic approach, i.e. on ontologies shared by the IoT community is being pushed forward within several Standard Defining Organisations such as ETSI M2M and OneM2M. One motivation of semantic abstraction resides in interacting with higher level entities rather than with sensors and actuators and thus making it possible to understand data without prior knowledge about their sources (device, web service,...)

2.3 Semantic Data Processing

Semantic web technologies allow logical reasoning so that new information or knowledge can be inferred from existing assertions and rules. IoT applications will require reasoning for various purposes such as resource discovery, data abstraction and knowledge extraction. To this purpose, specific algorithms are usually implemented within dedicated reasoners (e.g. Pellet, FACT++ and Jena) so developers do not need to be concerned with the complexities of the reasoning process itself. Examples of IoT resource discovery in the linked data can be found in (Pschorr et al., 2010).

We aim at applying this approach to integrating IoT Data and to experiment this approach in a real operational setting.

3 EXPERIMENTAL SETTING

As we have set high the ambition of assessing our approach in today's home, we have based our experimental platform on an off the shelf home automation solution called Homelive (hom,). Homelive allows people to manage their home appliances remotely. The Homelive pack offers a range of intelligent sensors and connected devices, brought by Orange's partners: weather monitors, thermostats, light switches, sound and movement detectors, water leak and smoke detectors, to name a few. We have thus instrumented

a space in our building with Homelive connected devices. It is worth noting that this space was already used by people for lunch around noon, coffee breaks in the morning, tea breaks in the afternoon, and for short breaks throughout the day during which people could engage in informal discussions or simply get some rest. Deploying Homelive in this space didn't have any impact on the way it was used. More specifically, we have installed 5 move detectors (entrance, kitchen, living and dining areas), 4 door state sensors (main entrance, fridge, freezer, medecine cabinet, drawer below the sink), 5 luminosity sensors (integrated in the move detectors), 5 thermometers (integrated in the move detectors) and 5 smart plugs (fridge, 2 coffee machines, boiler, TV set)

Each of these device is associated to a physical object to which it has been attached or into which it has been placed. For example, each electrical appliance including the fridge, the two coffee machine, a boiler and the TV set has been plugged into a smartplug, which itself is plugged onto the wall power socket. The number of some devices seem overdone, such as thermometers or the luminosity sensors, but actually, these sensors are integrated into the move detector device. Such devices are thus qualified as 3-in-1 which simply means that these physical device embed 3 different sensors.

Each device is assigned a name which makes explicit its type. Thus smart plugs have been named `MLPlug1`, `MLPlug2`, `MLPlug3`, `MLPlug4` and `MLPlug5`.

The picture displayed in Fig. 1 details where each device has been placed.

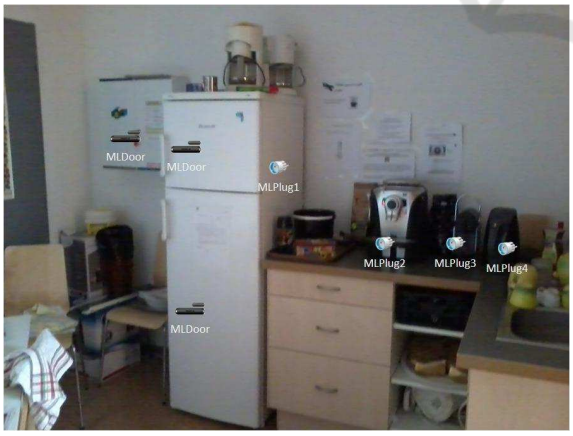

Figure 1: Devices deployment.

All these devices are connected through the wireless communication technology Z-Wave (zwa,). A Home Automation Box (HAB) is a dedicated gateway which makes it possible to access these devices from the IP world as depicted in Fig. 2, and make them part of the HAN (Home Area Network). In order to further extend their reachability from the HAN to the WAN (Wide Area Network), this HAB has to be connected to another gateway, such as the white box at the bottom of the figure. In our case it was an Orange Livebox.

Figure 2: Experimental setup.

In our experimental setup, the HAB collects all devices events and forwards them to a local server which will handle the aggregation and interpretation task.

Such device events are formatted in json, following a fixed "key-value" schema. An example of such an event emitted by smartplug `MLPlug1` is as shown in Fig. 3

From this comprehensive event description, which consists of 10 key-value pairs, we will mainly keep the following four:

`timestamp` is the date the event was received by the HAB. It is expressed as the number of seconds elapsed since jan. 1 ^{rst}, 1970 at 1:00AM.

`name` is the name of the device. As mentioned earlier we made is so that the type of the sensor could be identified from its name. For instance, we know from the name `MLPlug1` that the event has been emitted by a smartplug.

`variable` is the physical parameter that the event is about. In the event sample above, this parameter is the current electrical power consumed by the appliance it supplies.

```
{ "deviceId": "22",
  "deviceType": "BinaryLight",
  "homelive": "47122383",
  "id": "3",
  "name": "MLPlug1",
  "room": "A118",
  "service": "EnergyMetering1",
  "timestamp": "1428595051",
  "variable": "Watts",
  "value": "45" }
```

Figure 3: Event Description in JSON.

`value` is the current value of this physical parameter.

As you can notice, the above definitions of the keys are necessary for the reader to understand what the values associated to these keys mean, although the name of the keys have been chosen in a way that the reader would have figured out these definitions easily by himself. For an information processing system to correctly interpret an event, the meaning should be made explicit and even be embedded in the representation. In the section 4 we explain how we make this possible. But before that, in the following section we elaborate on why, remaining at the basic event description is too low a level to expect any interesting interpretation of the information it conveys.

4 SEMANTIC HOME DATA

IoT sensors are usually very talkative and versatile. For instance, in our Homelive platform, the smartplugs regularly delivers dense streams of power measurements that amount to up to 10 measures per minutes, although these smartplugs have been configured so that a new measurement is issued only if the current power consummed differs from the previous measure sent, by more than 10Watts. More generally Homelive devices send an event whenever a significant change in the data it measures occurs.

Such an abundance of information is superfluous to the inhabitants as well as to most smart home applications. These cumbersome data could be synthesized by applying one or more of the following policies:

- compute a mean value over a time slice of say 10mn
- compute a general trend from which strong increases and decreases could be easily detected
- or check compliancy to predetermined thresholds.

such abstraction of raw data will uplift the level of information and will place it closer to the home occupants'concerns.

The main idea is that we want to bridge the gap between low level raw data and high level information, so that the step that remains to be done to make a decision or to engage an action will be straightforward.

One positive side effect not to be underrated is that through this abstraction process, we reduce the size of the information and thus reduce the traffic. The gain in traffic size is particularly high if the this abstraction process is carried out close to the source of the information, i.e. close to the sensor. Having this process handled by the HAB or at least in the HAN is technically a reasonable solution.

In order for a computer system to be able to process this low level data, it is necessary to reformat this data into a representation that incorporates the semantics of the data as well as the data itself. Applied to the data produced by our homelive devices, this will result into a semantic model the SmartHome data. In the next section we explain this reformating process.

We first introduce the architecture of our system so that we get an overall perspective on where the raw data comes from, where the target semantic model will be stored and how it will be further exploited for high level interpretation and reasoning. This architecture is depicted on Fig. 4

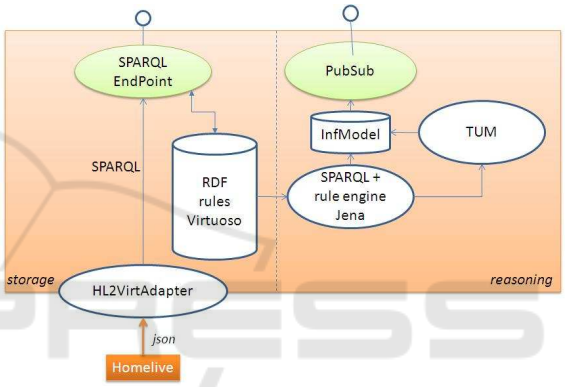

Figure 4: System architecture.

Input low level data is provided in a push/asynchronous mode by the Homelive HAB, that we have represented as a rectangular box on the lower left part of the diagram. As explained in section 3, this data consists of a flow of independent events emitted by each Homelive device. The HAB acts as a pass-through proxy which collects events from each device and forwards them right away to the local server, which has subscribed to receive such events as mentioned before. Events are described in json as shown in Fig. 3

Because we use the semantic web framework and its associated modeling languages RDF/OWL, our first task is to interpret the data conveyed in the event description in terms of elements of these languages.

An event is a piece of information that is produced by an IoT device. Thus we create a concept representing this piece of information and one representing this device. As this event is possibly not the first one produced by this device, the concept representing this device might already exists. In which case, we don't create it but will refer to the existing one instead, as will be shown later. Each key-value pair in this description

has to be properly annotated. Although those pairs are syntactically similar to each other, each of them express quite different things. For instance:

`"variable": "Watts"`
: means that the event reports about a physical phenomenon which is related to the rate at with electrical energy is consumed. This phenomenon is not specific to the event nor to the device that has produced this event. Thus we need to relate the information conveyed by this event to a concept that models this phenomenon. So, if this concept already exists we link the concept representing this information to the concept representing the phenomenon. If it doesn't exist we will simply create it. We call such concepts topics and create them as instances of a class called `InformationTopic`. The name of the link between Information instances and InformationTopic instances is called `isAboutTopic` The rate of electrical energy consumption is a quantitative characteristic of this phenomenon, which in the OWL modeling language we model as a datatype property link.

`"value": "45"`
: means that the target of this datatype property link should equate to the litteral value 45.

The process of semantically annotating the event description is illustrated in Fig. 5.

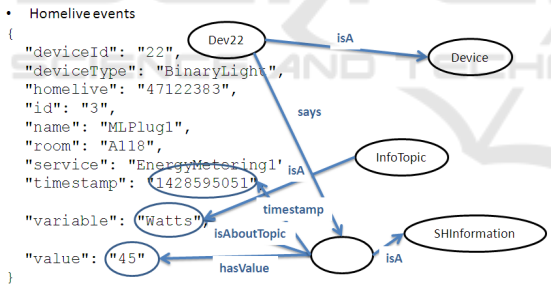

Figure 5: Homelive data semantic annotation.

As you see, some key-value pairs correspond to links between existing concepts, some refer to concepts that already exist or eventually that have to be created, some refer to litteral values to be assigned to concepts through links that already exist or eventually that have to be created.

Such analysis can be only conducted by one or a team of domain experts which collectively know the domain ontology, i.e. the catalog of concepts classes necessary to describe the application domain, the potential links between instances of these classes, and axioms that constrain the use of these links. An example of such an axiom is that the arity of the relation `hasValue` is 1, which means that a piece of information can only has one value and not more. Such axioms are necessary for the system to decide on the policy to adopt upon reception of new events from a device, which has already sent events about the same topic in the past. Note that this is generally the case, because once a device has been freshly provisionned and sent its first event to report about a physical phenomenon, its job is to update this report by sending other events. If the involved relations, such as `hasValue` is of arity 1 (or "is a functional relation" in the OWL terminology), the current target node in the model should be removed and replaced by the new node created by the event abstraction process.

The result of annotating one event description is a graph fragment consisting of a set of concepts which are classes of the ontology or instances of these classes, interrelated by relations of the ontologies. Some of these concepts are common to different events. Which means that assembling these fragments together will result into a larger graph which will aggregate and relate the information conveyed by the different event to each other. This graph gives an overall account of the state of the physical environement as seen by the pool of devices collectively. We call this state the situation. Then this graph constitutes a semantic model of the situation.

A userfriendly way to visualize this aggregated graph and more generally any RDF model, and to navigate along its edges is to use the Protégé editor (pro,). Using this editor, the semantic model of our SmartHome data can be displayed as shown in Fig. 6.

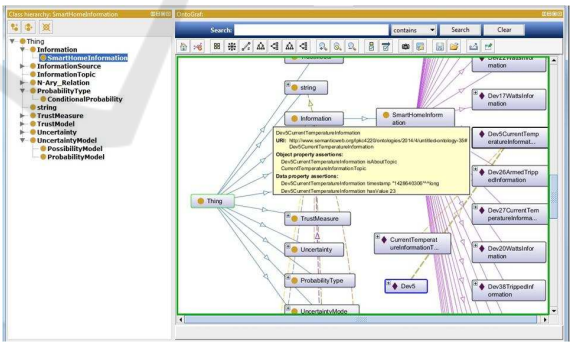

Figure 6: Browsing the model using Protégé editor.

The concepts classes of the ontology are displayed as a hierarchical tree on the left part of the screen. On the right part, concepts, links are displayed as a graph. Instances are displayed on the right. Hovering the mouse over nodes will popup datatype properties revealing the name of the datatype link and its litteral value. Hovering the mouse over links will reveal the name of the link.

In the following section we show how this situa-

tion semantic model can be easily exploited to infer higher level information, which can be directly processed to improve the home occupants experience.

5 DATA INTERPRETATION

Several inferences can be drawn on the basis of this semantic model. To give a glimpse of the variety we can mention the following three which have been identified has highly expected by endusers through dedicated focus groups conducted by the marketing department.

- Being informed that somebody has just entered the roomA118, can optimise your personnal productivity in case you have set up an appointement with a colleague for meeting at roomA118 and you don't want to wait unecessarily for her/his arrival.
- Infering that nobody's in the roomA118 is a useful information for those seeking quiet spots to shut themselves away and relax.
- Detecting that the fridge has being opened for more than one minute strongly suggest to have a look and close it if somebody has inadvertedly forgotten to close it. This will prevent damaging stored food and unecessary energy loss.

Let's now elaborate on how this inferences can be drawn. It would take too much space to detail the mechanism for all these inferences, so lets take the first inference "somebody has just entered" and elaborate this particular case.

Here is the basic reasoning: As shown in Fig. 1 a move detector `MLMove3` has been placed behind the door, and the door itself is equipped with a door opening detector `MLDoor1`. If a move has been detected by `MLMove3` after `MLDoor1` has detected that the door has been opened, for sure somebody has entered.

In order to check if the statement S:"`MLMove3` has detected a move after `MLDoor1` has been opened", we have to search the situation model for some fragment which describes this statement. Searching in a RDF model amounts to query it using SPARQL query language. A sparql query is a graph pattern, i.e. a subgraph defined using the same ontology elements (concepts and relations) than the complete graph but where some of the nodes, resp. some of the links, may be defined as variables, i.e. can match any node, resp. any link, in the graph. Before elaborating the SPARQL query, lets first define the graph pattern that describe statement S.

Whenever `MLMove3` detects a move it sends an event which as we have seen in section 4 updates a piece of information which captures information about movement within `MLMove3` detection zone. Information about mouvement is an instance of the class `TrippedInformationTopic`. We then have to search for a node which is an instance of `TrippedInformationTopic` and which is linked to the node that models `MLMove3` device with the relation `says`, as according to our ontology, this relation `says` links `Information` to its `InformationSource`. From this node we should search its value along the relation `hasValue` and its timestamp along the relation `timestamp`. This preliminary graph fragment, that describes this part of the search corresponds to the 6 nodes and corresponding links, on the upper part of the Fig. 7. Nodes which are searched are named with a prefix "?". For instance, the node representing the move information has been named `?info1`. This is also the case for `?move3ts` which represent the timestamp of the `?info1` information. We have to do the same for searching the status of the door informed by `MLDoor1` detector. This additional part of our search corresponds to the 6 nodes and corresponding links on the lower part of the figure.

Now that we've introduced the two timestamps `?move3ts` and `?door1ts`, our final question is "is `?move3ts` greater than `?door1ts`. If the answer is yes, then we can conclude that statement S is true. To complete our graph fragment, and thus add this last question to our SPARQL query, we will use a specific mechanism that the SPARQL language provides, for combining several variables and for defining "virtual" nodes, i.e. nodes which are defined in terms of other nodes of the graph fragment. In our particular case, we introduce the "virtual" node `?nbSecFromM3ToD1` which is computed by subtracting `?door1ts`' from `?move3ts`.

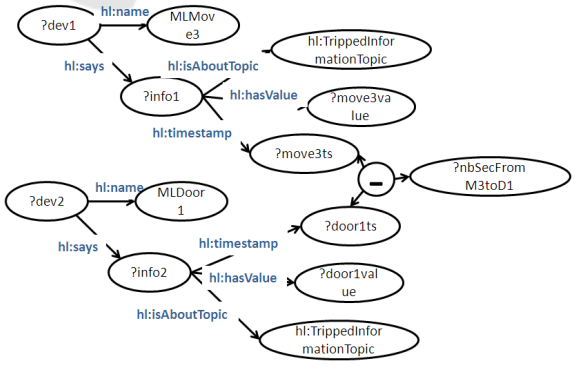

Figure 7: Has somebody moved in since the door has been closed?

Now that we've visualized the graph pattern that represents our query it is straightforward to format it using the SPARQL language. Fig. 8 shows how the

```
PREFIX hl:<http://www.orange.com/ontologies/shd#>
WITH GRAPH  <http://fiwarelod.orange-labs.fr>
SELECT ?m3val ?m3ts ?d1val ?d1ts ?nbSecFromM3toD1
WHERE {
?dev1 hl:says ?info1 . ?dev1 hl:name "MLMove3" .
?info1 hl:isAboutTopic hl:TrippedInformationTopic .
?info1 hl:hasValue ?m3val .
?info1 hl:timestamp ?m3ts .
?dev2 hl:says ?info2 . ?dev2 hl:name "MLDoor1" .
?info2 hl:isAboutTopic hl:TrippedInformationTopic .
?info2 hl:hasValue ?d1val .
?info2 hl:timestamp ?d1ts .
BIND((?d1ts - ?m3ts) AS ?nbSecFromM3toD1)}
```

Figure 8: SPARQL query.

query looks like.

Basically, each line in the `WHERE` section corresponds to an edge in the graph representation of the query as displayed in Fig. 7. Each line represents an edge as a triplet. For instance the line:

```
?dev1 hl:says ?info1 .
```

represents an edge where the source of the link has the identifier`?dev1`, the link has the identifier `hl:says` and the target of the link has the identifier `?info1`.

This work is ongoing. We have obtained good results on few experiments which show that inferences that we make with our approach are sound. However we plan to conduct an extensive testing campain that confirm the robustness of our system.

A wider perspective and discussion on these first results are developed in the next section.

6 CONCLUSION

Adopting semantic modeling technologies opens up an avenue of user experience improvements. For instance, one use case we've briefly evoked in our paper (Ramparany et al., 2014) but haven't tested with real data yet is to take into account information about devices location, such as rooms where the devices are located and devices functionality, such as the nature of data the device measures and reports in case it is a sensor. Aggregating devices information into the picture makes it possible for occupants to converse with their home with questions such as "what is the temperature in the kitchen?". This query would be decomposed into looking up all devices located in the kitchen (device location), then identifying which of those devices is a thermometer (device function) and finally retrieving the current temperature measured by this device. The answer to these 3 sub-query can be found in the aggregated RDF graph.

Widening the range of information sources beyond the IoT domain would even make possible fancier use cases. For example, if we don't limit ourselves to the restricted scope of smart home data, as we did in the work reported here, but aggregate data from the Open Data world, we could for example find out which IKEA cupboard would fit best in kitchen, in the space between the oven and the wall. For this, we simply need the dimensions of our kitchen and its appliances (our Smart Home data) and the dimensions of IKEA products. The laters could be found on the IKEA online catalog if this catalog is available as open data. Once aggregated on the common semantic model, the respective dimensions could be compared.

To this view of having IoT system access open data sources, there's of course the dual view point, of inserting the IoT in the realm of the semantic web and consider our home, our car, the city as new contributors to the semantic web by having them publish real-time information about themselves, their states, their moods, etc... By the end of the day, the philosophy would be the same: an aggregation of data originating from the IoT and the one side and of data from the public Web on the other side, to form a consolidated model, and reason upon this consolidated model. The main difference is about where the aggregation takes place and who performs the reasoning, an IoT application or a Web service? Who cares? the technology to implement these processes would probably be the same, and as attested by our experiments this technology is there and mature enough to be applied.

REFERENCES

Homelive: Confort et domotique, maison connectée. http://homelive.orange.fr.

Protégé: A free, open-source ontology editor and framework for building intelligent systems. http://protege.stanford.edu/.

Z-Wave: Home control. http://www.z-wave.com.

(2015). http://www.computerweekly.com/news/2240217788/Data-set-to-grow-10-fold-by-2020-as-internet-of-things-takes-off.

Berners-Lee, T. (2006). "linked data". In *International Journal on Semantic Web and Information Systems*, volume 4. W3C.

Bizer, C., T., H., and Berners-Lee, T. (2009). "linked data - the story so far". In *International Journal on Semantic Web and Information Systems*, volume 5.

Calvanese, D., Giacomo, G. D., et al. (2011). The mastro system for ontology-based data access. In *Semantic Web Journal*.

Compton, M., Barnaghi, P., et al. (2012). 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.

Hartig, O., Bizer, C., et al. (2009). Executing sparql queries over the web of linked data. In *The Semantic Web-ISWC*, pages 293–309. Springer Berlin Heidelberg, W3C Working Group.

Lenzerini, M. (2011). Ontology-based data management. In *Proc. of CIKM 2011*, pages 5–6.

Pschorr, J., Henson, C., et al. (2010). Sensor discovery on linked data. *Proceedings of the 7th Extended Semantic Web Conference, ESWC2010, Heraklion, Greece*, 30.

Ramparany, F., Marquez, F. G., et al. (2014). "handling smart environment devices, data and services at the semantic level with the fi-ware core platform". In *Proceeding of the 1st Workshop on Semantics for Big Data on the Internet of Things (SemBIoT 2014)*, pages 14–20. IEEE International Conference on Big Data.

Ramparany, F., Poortinga, R., et al. (2007). An open Context Information Management Infrastructure - the IST-Amigo Project. In of Engineering, I. I. and Technology, editors, *Proceedings of the 3rd IET International Conference on Intelligent Environments (IE'07)*, pages 398–403, Germany. University of Ulm.

Rodriguez-Muro, M., Hardi, J., et al. (2012). Quest: Efficient sparql-to-sql for rdf and ow. In *Proc. of the 12th Int. Semantic Web Conference (ISWC 2012)*.

Sorici, A., Picard, G., et al. (April 2015). CONSERT: Applying semantic web technologies to context modeling in ambient intelligence. In *Computers & Electrical Engineering*, number 44.

W3C. Web ontology language. Technical report, W3C, http://www.w3.org/TR/owl2-overview.

Ye, J., Dasiopoulou, S., et al. (2015). Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing*, 23:1 – 25.