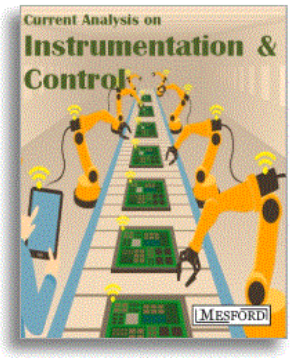

Using Semantic Web Technologies for Smarter and Sensible Homes

Fano Ramparany*, Imane El Arabi, and Mondi Ravi

Orange Labs, 28 chemin du Vieux Chêne, 38240, Meylan, France

Abstract:

In smart environments, devices native languages are machine-understandable and consist of low level messages such as raw data measurements reported by sensors or commands sent to actuators. Conversely, people living in these smart environments expect to interact with devices through higher level concepts such as "the presence of people in a room", or the "desired luminosity of a room". Making everyday objects understand and satisfy people desires requires mechanisms for uplifting these objects' languages to the level of people languages and conversely to transcode people wishes into everyday objects' application programming interface (API) messages. From now on, we refer to these everyday objects as "Things" and we refer to this gap between things languages and people languages as the Internet of Thing (IoT) semantic gap. In parallel, knowing people availability and interests also makes it possible to confront them to Open Data resources. This opens up the possibility to find out which resources best satisfy them. In this article we demonstrate how we have used semantic web technologies to implement innovative smart home services. We focus our description on two key capabilities inherent to those technologies:

- Their ability to help bridge the IoT semantic gap in a standardized and principled way.
- Their potential for interoperability with public Open Data.

We illustrate our approach with typical use cases in the domain of home automation such as presence detection and light monitoring, and social networking such as gathering recommendation. We also present implementations of these use cases and their validation on real data.

Publication History: Received: 30 November 2018| Revised: 23 January 2019 |Accepted: 24 January 2019

Keywords:
Semantic web, Internet of Things, Open Data, Linked Data, Reasoning, SmartHomes.

1. INTRODUCTION

Today, Smart Environments (SE) refer to places of our everyday life, which are equipped with connected devices such as sensors and actuators. The intelligence of SEs amounts to observe people's activities through sensors measuring data produced by these activities (e.g. a temperature rising, a door opening, etc.) and to derive specific modalities of actuators behavior (e.g. switching on/off lights), so that these actuators, and thus the environment that these actuators literally animate, adapt to people activities in real-time.

Much work has addressed this issue, specially under the scientific domain of "Context Awareness". Context awareness is considered as a key technology within the IT industry, for its potential to enhance user experience and to provide a major differentiation among service providers. According to a Gartner Inc. report [11], "Context-aware computing today stands where search engines and the web did in 1990".

One major challenge stems from the conceptual gap that exists between the languages of devices and things on one side and the languages of people on the other side. The native language of "Things" consists of low level messages such as raw data measurements reported by sensors (e.g. 70°C reported by Thermometer45) or command instructions sent to actuators (e.g. SWITCH-OFF command sent to Smartplug3). On the other side, people living in these smart environments expect to interact with things through higher level concepts such as "the presence of people in a room", or the "desired luminosity of a room".

*Address correspondence to this author at Orange Labs, 28 chemin du Vieux Chêne, 38240, Meylan, France; E-mail: fano.ramparany@orange.com

Mesford Publisher Inc
Office Address: Suite 2205, 350 Webb Drive, Mississauga, ON L5B3W4, Canada; T: +1 (647) 7109849 |E: caic@mesford.ca, contact@mesford.ca, https://mesford.ca/journals/caic/

Making things populating smart environments understand people' desires requires mechanisms for uplifting the languages of things to the level of people languages and conversely to transcode people wishes into "Things" application programming interface (API) messages. We refer to this gap between things languages and people languages as the Internet of Thing (IoT) semantic gap.

Early work has addressed this issue using ad hoc approaches with very limited genericity. The introduction of semantic modeling techniques and languages such as RDF, RDFS and OWL to express sensors data has improved the interoperability of this data and improved its ability to characterize specific contexts ([7], [20]). Thus these contexts can be shared among use cases and lead to better genericity in the solutions. However information still remains at too low a level, i.e. too close to sensors' languages. The introduction of reasoning mechanisms such as Answer Set Programming ([12], [14]) and rule based inferencing ([16]) has partly filled in this semantic gap, especially on the question of interpreting sensors low level data in terms of human concepts.

However, to our knowledge none of these works has bridged the gap when it comes to getting actuators understand human concepts.

In this article we propose a complete solution based on semantic web standard technologies including OWL, SPARQL, SWRL and IoT domain specific ontologies that bridges the IoT semantic gap, along both direction and in a standard and principled way.

The paper is organized as follows: In the next section we introduce some use cases that illustrates the IoT semantic gap issue. We then introduce the experimental platform that we used to experiment and assess our solution. This enables us to illustrate the technical and scientific challenges that we face with a real Smart Home setting. We then present our solution to address this issue. We finally describe the first implementation of our solution and report on the results obtained by experimenting it on real data.

2. IOT SEMANTIC GAP USE CASES

People do care about their home, where they spend more than half of their lifetime [5]. When they are away from home they are concerned about the comfort and protection of their family who are still there or by keeping their home safe from burglary and any natural disaster. Homes take an important place in people's heart and people share strong affective links with their homes. Recent studies have investigated the use of social network media such as Facebook, Twitter to establish a privileged connection between people and their home, like they do for their friends. Thus homes could chat, tweet and send selfies to keep people reassured about their home and its occupants [10].

In our work, we investigate mechanisms and strategies to relate high level concepts such "the home is empty", "most people are sleeping", "the home is being breached" and "light the corridor", from low device level data and commands such as "move detection", "temperature" and "switch on/off".

The paper makes use of four use cases. The first three ones have deliberately been kept simple, for pedagogical purposes. However, these simple use cases we provide some hints on how to handle more complex situations and tasks using the same approach and methodology.

The first use case deals with inferring presence in a room based on raw data produced by home automation sensors deployed in the room. More specifically, our Home friend tells us about the number of people it hosts. The possible outcomes are : "there is nobody here", "there is at least one person here" and "there are more than one person here". This information could be provided by the "Home" friend if we send it a text message asking "is there somebody at home?"

The second use case deals with detecting and notifying a misusage of the fridge. It handles the situation where the fridge's door remains open for more than a minute. Once detected, this abnormal situation could be displayed on a virtual wall such as the status area of a Facebook page.

The third use case deals with context-aware lighting. In this use case the main issue is to be able to automatically switch on the right lamps based on people location. In contrast to the first two use cases, this one involves actions and device activation.

In the fourth use case we aggregate home automation data and context with open public data such as the Electronic Program Guide content, in order to foster social interaction through a dedicated social network application.

To implement these use cases, we apply a rule based reasoning method on a description of the physical environment that is built and maintained by an enabler called FLOD, which stands for "Future Internet Linked Open Data Enabler"..

3. SMART HOME SETTING

We have based our experimental platform on an off-the-shelf home automation solution called Homelive [3]. 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 but a few.

We have instrumented a space in our building with Homelive connected devices. It is worth noting that this space was already used by people for lunching around noon or having coffee or tea breaks throughout the day, while they could engage in informal discussions or simply relax. Deploying Homelive in this space did not have any impact on the way it was used.

Each Homelive device forms a context source and is assigned a name which makes explicit its type. Thus smart plugs have been named MLPlug1, MLPlug2, MLPlug3, MLPlug4 and MLPlug5. They send data related to the electrical consumption by the different appliances connected to them.

The pictures displayed in Fig. 1 detail where each device has been placed.

Fig. (1). Devices deployment.

All these devices are wirelessly connected through the wireless communication technology Z-Wave [4]. 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 an internet gateway, such as the white box at the bottom of the figure. In our case it was an Orange Livebox.

The process of translating this JSON representation into RDF has been described in detail in a previous article [15].

In the following section, we introduce FLOD.

4. RELATED WORK

The task of inferring high level information from low level data has been investigated in the field of Artificial Intelligence for a long time and is still an active research topic today. Iconic instances of that problem solving task include:

- diagnosis in the medical domain [23], where a disease has to be identified based on some observed symptoms
- image understanding, where a person has to be identified from a still image, or a human activity has to be recognized from a video sequence [13].

There are alternative approaches for solving these types of problem solving task. To the ones that are based on neural networks, and its popular variant called "deep learning", we prefer a symbolic approach where inferences are based on domain expertise rules that are triggered by a dedicated rule engine. We have made this choice because the NN approach requires a preliminary stage of supervised learning, which itself requires a lot of sensor data to be collected, stored and properly formatted and manually labelled with the different interpretations to be inferred.

5. APPROACH AND ARCHITECTURE

We have based our development on a Semantic Context Management Service (CMS) [18]. The CMS is an open infrastructure for managing context information. Its role is to acquire information coming from various sources, such as physical sensors, user activities, and applications in process, or internet applications and to subsequently combine or abstract these pieces of information into "context information" to be provided to context aware services. Its most salient features include:

- It's compliance to the web service architecture both for interfacing to context consuming applications and

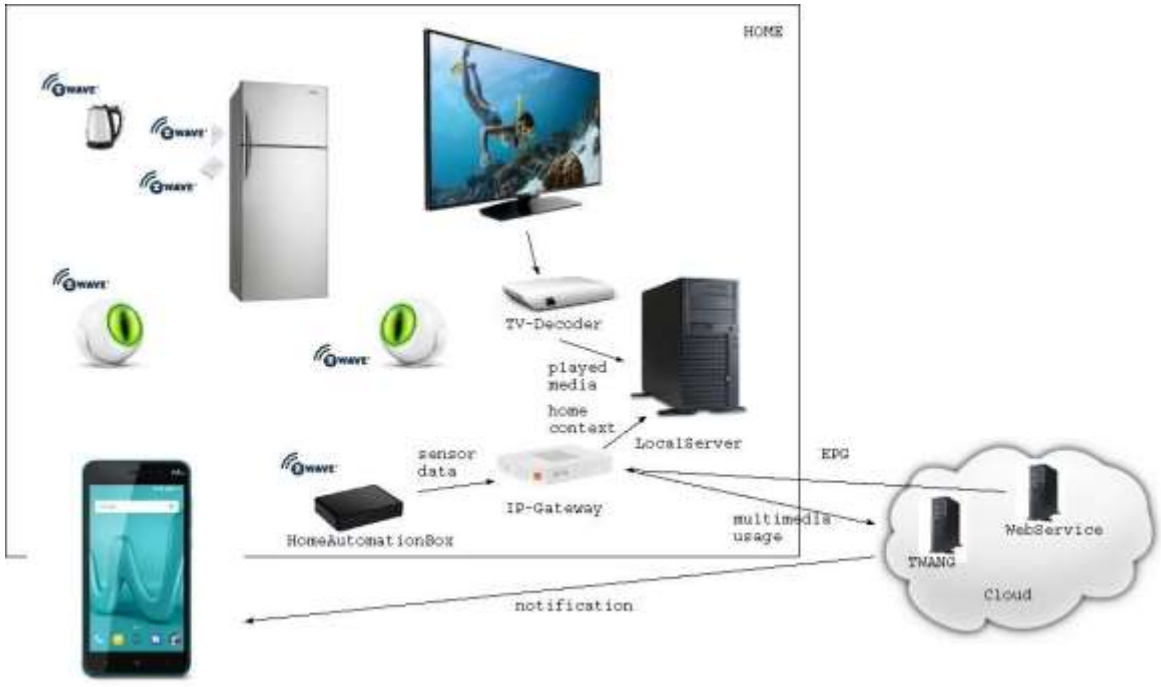

Fig. (2). Experimentation platform.

for integrating its sub components.

- The modeling of context information using a high level language with much expressiveness
- state of the art context sources such as a context history manager and an an audio based positioning system

This CMS has been adapted to the IoT domain through the FLOD semantic broker enabler [17]. FLOD has kept the same philosophy as the CMS. More specifically as depicted in figure 3, it serves as a mediation platform between context sources which are entities that provide context information and context aware application. Context aware applications are applications that benefit from context information for adapting their behavior. Examples of such context aware applications are Process Aware Information Systems (PAIS), Ontology-based Information and Extraction systems (OBIE), Process Querying. This mediation platform implements and provides functions for collecting rough data from context sensors, for formatting these data in RDF and aggregating them with previously collected data. This aggregation results into a semantic context model. This context model can then be queried by context aware applications through the standard SPARQL query language, or processed by reasoning mechanisms. FLOD inherits its reasoning capabilities from its using of semantic modeling techniques. One particular policy that FLOD adopts is to ensure that the RDF documents contained in its context model comply to well defined ontologies.

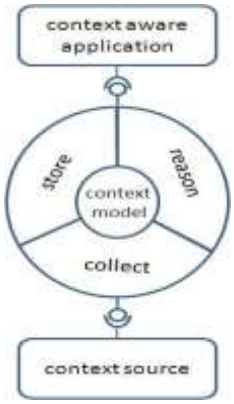

Fig. (3). FLOD Semantic Context Broker functional architecture

As prescribed by cognitive engineering methodologies and good practices, we have looked up existing ontologies that cover our universe of discourse, i.e. the scope of concepts and issues that we need to model in our application. We ended up using:

- the IoTA ontology [7] to model IoT devices and data measurements.
- the SAREF ontology [6] to model Smart appliances API

We indirectly use some higher level ontologies such as the time ontology [8] and the WGS84 (positioning and geolocation) ontology [2], because they are themselves imported by the IoTA and SAREF ontologies.

We have also defined our own planning ontology for modeling and reasoning about actions and how actions change the state of affairs. Concepts of this ontology are needed to implement the context-aware lighting, as the ultimate goal here is to activate the right lamps. Previous work has addressed the issue of semantically modeling the generic domain of planning and action, as in the Process Planning Knowledge Model [9]. Among them, some have also used OWL as the modeling language. However, their model was only targeted towards the domain of workflows construction.

In the diagram 4 we depict some concepts of the integrated ontology. These are the main classes that we used from these ontologies, together with their interrelationships.

For instance, the classes "Device", "Function" and "Command" come from the SAREF ontology, whereas the classes "Goal" and "State" come from the action and planning ontology.

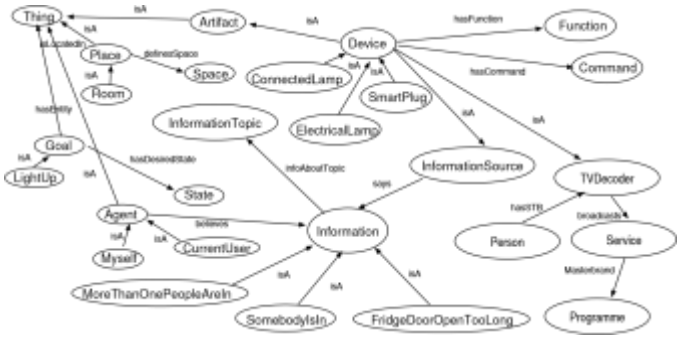

Fig. (4). FLOD ontology.

The system has its own representation of the world that we represent as a semantic model, i.e. as a set of RDF triples where the first and third element are instances of FLOD classes. Each triple denotes a piece of information.

In section 6 we illustrate how our extended enabler has been used to implement a system which solves the scenarii described in section 2.

6. HOME MOOD SCENARIOS IMPLEMENTATION AND EXPERIMENTATION

In this section we describe the FLOD system in terms of software architecture and its main components. Figure 5 presents our IoT context broker FLOD architecture.

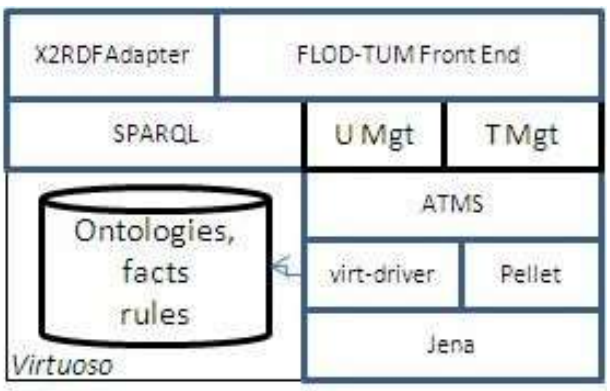

Fig. (5). FLOD Semantic Context Broker system architecture.

We introduce each of the software component and its role in the following paragraphs. In order to describe the interface of FLOD to its physical environment, we have included in our description, real instances of context sources (as defined in section 5 and depicted in figure 3).

6.1. Data Collection and Preprocessing

The HAB collects all devices events and forwards them to the X2RDFAdapter component (see figure 5) which will handle the formatting of the event into RDF. Such device events are formatted in JSON, following a fixed "key-value" schema.

6.2. Presence Inference

In our experimental setting, there is a unique entry door to access the room. This door has an automatic door closer. This configuration enables us to postulate that if a sign of activity is detected in the room after the door has been closed we can infer that there is somebody in the room. There are several such signs including, the opening or closing of a drawer or one of the two fridge doors, the detection of movement reported by any of the five move detectors and the switching on or off of the TV or of the boiler.

We model this reasoning process by defining SWRL [22] rules that infer a sign of activity for each possible cause. Then we have a rule that infer presence from a comparison between the last activity reported and the last entry door closing. As explained earlier if the last activity is anterior to the entry door closing, the conclusion part of the rule specifies that there is someone in the room. We show one of the activity reporting as well as the presence inference rules in the SWRL syntax below:

isAboutTopic(?info1,DetectionTopic) &

says(?dev, ?info1) &

MotionSensor(?dev) &

hasValue(?info1, 1)

-> believes(MySelf,SomebodyIsIn)

In order to infer the presence of more than one person, we detect the simultaneity of activities occurring in different places. For instance, if one fridge door is open at the same time than a move is detected in the living room area, one single person cannot be responsible for these two activities. We can conclude that there are at least two persons in the room.

We show one of the activities simultaneity detection rule below:

isAboutTopic(?info1, DetectionTopic) &

isAboutTopic(?info2, DetectionTopic) &

says(?dev1, ?info1) &

says(?dev2, ?info2) &

MotionSensor(?dev1) &

MotionSensor(?dev2) &

diff(?dev1, ?dev2) &

hasValue(?info1, 1) &

hasValue(?info2, 1)

-> believes(MySelf,MoreThanOnePeopleAreIn)

These SWRL rules are evaluated and processed by the Pellet reasoning system [19]. Pellet requires Jena library and its virt-driver driver to be able to communicate with Virtuoso

Once the presence of one, or two or more people is inferred this presence information persists until the entry door is opened again. Once the door is opened, this presence information has to be removed because the people inside could have exited the room while the door was open.

The overall RDF graph forms the context model, which any third party application could access through a REST API exposed by the FLODTUM-FrontEnd component. This API has been used by a simple client application called HomeMonitor that we have specifically developed to test the FLOD enabler. This experimentation has also been carried out in order to assess FLOD reliability when applied to the Home mood scenario described in the previous section. In the next section, we introduce this HomeMonitor client, the testing experiments and the results obtained.

5.3. Fridge Misusage Alert

This use case, like the previous use case "Presence inference" belongs to the same problem solving class which is "scene interpretation" Scene interpretation. We talked about that type of problem solving task at the beginning of section 4. In both use cases the task is to analyze low level data provided by the various sensors and detect and recognize a specific configuration of these data that corresponds to a context that we are interested in. In contrast to the "Presence inference use case", the "Fridge misusage" use case requires a more sophisticated analysis of the temporal properties of event data produced by the "door opening" sensor that instruments the fridge door.

We also use the condition part of a SWRL rule to specify the configuration of the "door opening" sensor data that corresponds to the situation or context where the fridge door has remained opened too long. This rule is defined as follows:

DoorSensor(?dev) &

monitors(?dev, ?artifact) &

Fridge(?artifact) &

says(?dev,?info) &

isAboutTopic(?info, DetectionTopic) &

hasValue(?info,1) &

timestamp(?info, ?tsFridgeOpen) &

timestamp(FLODInstantNow,?tsNow) &

subtract(?nbSecFridgeOpen, ?tsNow, ?tsFridgeOpen) &

greaterThan(?nbSecFridgeOpen, 60)

-> believes(MySelf, FridgeDoorOpenTooLong)

Whereas the condition part of the SWRL will recognize this context, the detection of this context will be handled by the rule engine which will constantly monitor the model and trigger this rule whenever the condition part is satisfied. The condition part is slightly more complex than that of the "Presence inference" use case as some test has to be performed on the event timestamp. For instance, the last two conditions:

subtract(?nbSecFridgeOpen,?tsNow, ?tsFridgeOpen) &

greaterThan(?nbSecFridgeOpen, 60)

check whether the current time ?tsNow exceeds by more than 60sec the last time ?tsFridgeOpen that the fridge door has been opened.

The action part will then assert this specific context FridgeDoorOpenTooLong in the model.

5.4. Lighting Control

In this use case, it is dusk time and one inhabitant expresses his wish to enlighten the place he is in. A natural and user friendly way for that person to express his wish is through a voice based user interface. We assume that the speaker is recognized and that the voice message content is understood. Today, voice technology is mature enough to ensure that our assumption is reasonable.

Thus, the reasoning tasks to be performed are the following:

- use the person location to detect which room to light up
- identify the relevant lamps, i.e. the lamps that are located in the room where the person is standing
- identify the commands that should be sent to the lamps found at the previous stage

Again, those tasks are implemented using simple SWRL rules. We detail each of these rules. The first two rules make it possible to retrieve things that are useful to light a place.

ConnectedLamp(?x) -> LightingThing(?x)

ElectricalLamp(?x) -> LightingThing(?x)

The next rule handles the first and second task. They assert that if the user needs light, the room he is standing in should be lighted:

LightUp(?goal) &

CurrentUser(?user) &

isLocatedIn(?user, ?space) &

definesSpace(?place, ?space)

->

hasEntity(CurrentGoalSES, ?place) &

hasDesiredState(CurrentGoalSES, Lighted)

The following three rules handle the last task. One asserts that to set a place to the state "Lighted", lighting artifacts located in that place should be activated.

hasEntity(?goal, ?place) &

hasDesiredState(?goal, Lighted) &

LightingThing(?artifact) &

definesSpace(?place, ?space) &

isLocatedIn(?artifact, ?space)

->

ActivatesArtifact(?artifact)

The next one tells how to activate a standard ICT equipment, i.e. a connected equipment.

ActivatesArtifact(?artifact) &

hasFunction(?artifact,?artifactFunction) &

hasCommand(?artifactFunction,?command) &

OnCommand(?command)

->

sendsCommandTo(?command,?artifact)

This last rule tells how to activate an non connected electrical equipment plugged into a smart plug

ActivatesArtifact(?artifact) &

isElectricallyPluggedInto(?artifact, ?device) &

SmartPlug(?device) &

hasFunction(?device,?function) &

hasCommand(?function,?command) &

OnCommand(?command)

-> sendsCommandTo(?command,?device)

In order to identify the devices to control, and to determine the exact commands to send them, we submit the following query:

PREFIX flod: <http://orange.com/flod.owl#>

SELECT ?device ?command

WHERE {

?command flod:sendsCommandTo ?device

}

The response to this query will contain a list of bindings (commandi,devicei), where commandi is an instance of the class Command, and devicei is an instance of the class Device. These bindings state that to achieve the current LightUp goal, the command commandi should be sent to each device devicei, that is mentioned in the response.

5.5. TV Based Social Interaction

For this use case, we have implemented a service named TWANG. This service uses the electronic program guide (EPG) [1] as a source of Open Data and focuses on the television usage in particular. It also deals with some social issues by encouraging neighbors to do activities together.

More specifically, each member of a given neighborhood creates their profile (personal information, list of friends, programs for which they agree to publish viewing information and preferences in terms of themes e.g. sports, talk shows, etc).

During the nominal operation of TWANG, the TV decoder transmits in real time data from the TV to the TWANG server, such as the TV program viewed by the users or the electronic program guide.

The server collects all this data and looks for matches that could lead to a grouping of neighbors. For example, if two people are watching the same TV show, and the theme of this show is one of the favorite themes of each person, then they represent a good match. When such correspondence is established, a matching notification is sent to a first person, suggesting that they invite the second person to join them to participate together in this activity. If the first person accepts, an invitation is sent to the second person.

TWANG deployment architecture is represented in the figure 2. To model the context, we relied on the ontologies design best practices. So we started by looking for existing ontologies that are related to our fields of interest including the description of people, the internet of things and TV programs. We then used some of the most known a reliable ones that contain relevant concepts such as: FOAF (”Friend of a Friend”) for describing persons, their activities and their relations to other people and objects. The Programs ontology developed by the BBC which is a vocabulary for describing programs and that defines concepts such as brands, series, episodes, broadcasts, etc. SAREF, Smart Appliances REFerence ontology, that aims at representing functions of devices that are present in households, public buildings and offices.

We have imported those existing ontologies in a new ontology that we named ”twang”. We then added the missing entities and relations to it. In our ontology we created a new class named STB that represents the set up box, to which we have added a mac address data property, and two object properties that relate the STB with its owner and the service that it broadcasts. Figure 4 on the right shows how we modeled the data stored by the system. Data about the users is entered via a web interface, where there is a form dedicated to personal information, preferences and neighbor list. The open data source we used was exported from a web service in XML format. It contains information about every program such as its title, summary, and its duration.

In order to find users who represent a good match, we run a script in the server. This script implements a loop that queries the RDF graph, taking into account all the parameters that we introduced earlier. If a match is found, it sends a notification to the corresponding users.

```
PREFIX tw:<http://ontology.orange.com/twang#>
PREFIX dcterms:<http://purl.org/dc/terms/>
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX foaf:<http://xmlns.com/foaf/0.1>
PREFIX po:<http://purl.org/ontology/po/>
WITH GRAPH <".$GRAPHURI.">
SELECT ?pName ?spName ?token ?stoken ?serviceDesc ?duration ?sphone ?phone
WHERE {
?p foaf:based_near ?sp .
?p tw:hasSTB ?STB .
?p foaf:name ?pName .
?sp foaf:name ?spName .
?p tw:hasPhoneNumber ?phone .
?sp tw:hasPhoneNumber ?sphone .
?sp tw:hasSTB ?sSTB .
?STB tw:broadcasts ?service .
?sSTB tw:broadcasts ?service .
?p tw:hasPhonetoken ?token .
?sp tw:hasPhonetoken ?stoken .
?service dc:description ?serviceDesc .
?prog po:masterbrand ?service .
?prog po:duration ?duration .
}.
```

Figure 6 shows the content of the notification received by the users on their phones. It displays the name of the neighbor who is matched, the name of the channel they are watching together and also how much time is left before the show is over. The user can then either ignore the notification, or ask their neighbor to meet.

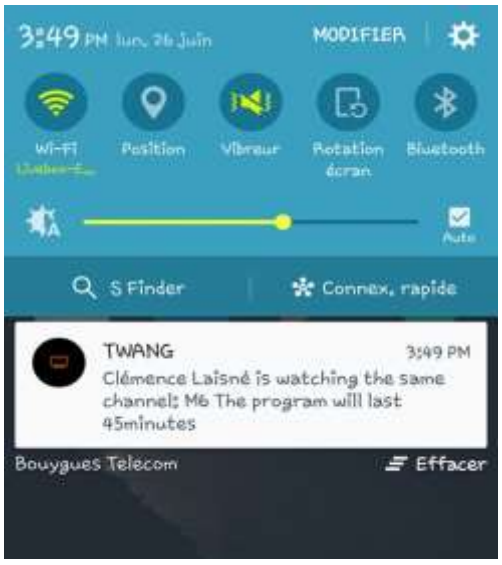

Fig. (6). The notification message.

7. TEST AND RESULTS

We have developed HomeMonitor, a simple FLOD client application to assess the reliability of our system.

HomeMonitor application connects to FLOD and asks about which beliefs FLOD has on the home moods. This query will trigger the Pellet reasoner which will in turn evaluate and eventually run the SWRL rules introduced in section 6.2 for the Presence inference use case.

The following table **1** provides a quantitative evaluation of the results of this comparison. For example, for each actual presence situation (for instance, when there is one person in the room, i.e. on the second line of the table), we report the presence information inferred by FLOD (for instance, "nobody is in", i.e. the first column, "someone is in", i.e. the second column and "more than one person are in", i.e. the third column). The value mentioned in one cell is the cumulative duration of the situations where FLOD has inferred the presence information that corresponds to the column. For example, for the total duration of the experiment, if we consider the cumulative time during which there was only one person in the room, during 90% of this time FLOD has yield the correct inference (i.e. there is someone in the room), during 6% of this time, FLOD has inferred that nobody was present in the room, and during the remaining 4% of this time, FLOD has inferred that there were at least 2 persons in the room.

As we can see, FLOD is pretty accurate because the scores on the diagonal are very close to 1. There is an exception when there are more than one person present (at the 3 rd row and 3 rd column), but we know that this has more to do with the condition we have set for inference that there are more than one person, than with the inferring process itself. Actually, if two or more people enter in the room at the same time and stay close to each other for a while, the rule for inferring the presence of more than one person will not trigger until the group split and at least two persons are close to two distant move sensors simultaneously. This might take a while and until the group split occurs, the system will consider that there is one person.

FLOD is also pretty robust, because it never infers that there is more than one person while there is nobody in the room, nor the other way around.

Table 1. Interpretation Confusion Matrix.

Real \ Inferred	Nobody	Someone	More Than One
Nobody	0.99	0.01	0.0
Someone	0.06	0.9	0.04
More than one	0.0	0.25	0.75

The use case Fridge misusage alert corresponds to a situation that occurs rarely and the Lighting control correspond to a sporadic one. Thus we haven't step up a dedicated extensive testing campaign to assess the validity of our system. Instead we have verified that an alert was correctly sent at the right time whenever the fridge door was opened and that the correct lamps were identified for lighting the room where the person asking for light was standing.

We have also conducted a performance evaluation of the use case TV based social interaction under different load conditions. By varying the number of users, we have analyzed how the time needed to find matchings how the number of triples stored in the database vary.

We have created a graph to store the ontology we are using to model data, and another graph just for the instances. To start with, we have uploaded to the instances graph data about the TV programs. It concerns programs broadcasted by 101 channels (one random program per channel during a given day). Then, we added progressively 50 different user profiles by filling the user form we implemented. The figure 7 the number of users we added each time, the corresponding number of triples generated and the average response time in milliseconds.

Number of persons	3	10	20	30	40	50
Number of triples	886	979	1149	1326	1490	1654
Average response time (ms)	337,68	343,73	344,73	345,84	344,47	440,52

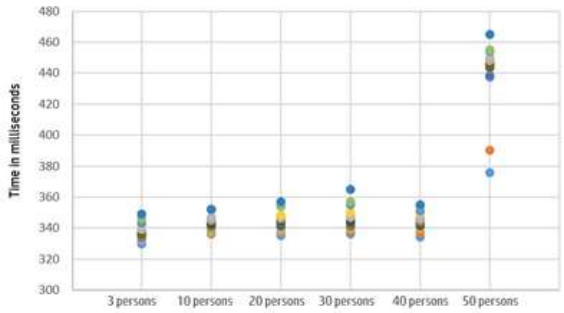

Fig. (7). Evaluation of TWANG

As expected, when the number of persons increases, the number of triples increases too. From 3 persons to 50, the corresponding number of triples goes from 886 to 1654. Concerning the average response time, it does not vary much when the number of users is less than 40 persons. In fact, it goes approximatively from 337 to 345 milliseconds. But when the number of users is 50, the average response time increased by 100 milliseconds.

The average response time depends on many factors such as the complexity of the query applied, the number of triples in the graph, the robustness of the triple store and the capacity of the machine or machines used etc.

Since we are using FLOD as a triple store which is based on Virtuoso, we knew in advance that it can handle large numbers of triples [21] and that it had good performances compared to other open source products.

Using a semantic web based approach allowed us to make explicit the relationships that exist between the different models we are handling in the use cases and create a continuity between different sources of data since the instances were stored in the same graph using one data model that imports other ones. Having a RDF data model is easy to query and makes it possible to apply a reasoner on the instantiated graph and deduce new information on the fly. This approach turned out to be fruitful since the performance results were very promising and the implementation fulfilled all the specifications of the use cases.

8. CONCLUSION AND PERSPECTIVES

In this paper we have addressed the issue of interpreting and controlling smart environments at a high conceptual level. Our solution makes it possible to the people to interact with their environment using their natural and day-to-day vocabularies without referring to technical details, such as sensors and actuators, their related functionalities or location.

We have proposed a novel framework which is based on the following elements:

- the combination of ontologies covering the domains of IoT devices, smart appliances, data measurement, context modeling, positioning and geolocation, time, and planning and action, multimedia content.
- the use of rule based reasoning to interpret low level data measurements statement into higher level context statement and to interpret higher level user's goals into low level device control instructions.

We have implemented this framework as a generic enabler called FLOD. This enabler has been tested on real data in the framework of a smart home application. Within this application, FLOD has been used to infer human presence context in a room instrumented with low level sensors.

Our first experiment has shown that FLOD inferences are both accurate and robust. We plan to consolidate these promising results through an extensive experimentation campaign. We have shown that a large range of smart home contexts can be easily implemented with our approach, including the following situations: "the coffee machine has been used less than 1mn ago" and "the light is on for no reason". This will greatly improve our experience when interacting with our home.

We have shown that our approach can be applied to more complex use cases as well. For instance, we have developed the TWANG social network service which can increase users sociability strengthen their relationships with their neighbors. It can also be way of saving the energy by reducing the number of screens turned on, and it will help freeing the shared equipment for other members of the family to use.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1]. EPG definition. Available from: https://en.wikipedia.org/wiki/Electronicprogramguide

[2]. Basic geo vocabulary. Available from: https://www.w3.org/2003/01/geo/ (2004)

[3]. Homelive: Confort et domotique, maison connectée. Available from : http://homelive.orange.fr (2017)´

[4]. Z-Wave: Home control. Available from: http://www.z-wave.com (2017)

[5]. ATUS: Available from: http://www.bls.gov/tus/charts/#about (2017)

[6]. Daniele L, den Hartog FTH, Roes J. Created in close interaction with the industry: The smart appliances reference (SAREF) ontology. In: Formal Ontologies Meet Industry - 7th International Workshop, FOMI 2015, Berlin, Germany, August 5, 2015, Proceedings. pp. 100–112 (2015)

[7]. De S, Barnaghi P, Bauer M, Meissner S. Service modelling for the internet of things. In: proceedings of the 3rd Workshop on Software Services (2011)

[8]. Hobbs J, Pan F. Time ontology in owl; 2006. Available from : https://www.w3.org/TR/2006/WD-owl-time20060927/

[9]. Huang Z, Qiao L, Anwer N, Mo Y. Ontology model for assembly process planning knowledge. In: 21st International Conference on Industrial Engineering and Engineering Management 2014. Zhuhai, China

[10]. Kamilaris A, Pitsillides A. Social networking of the smart home. In: Personal Indoor and Mobile Radio Communications. pp. 2632–2637. IEEE 21st International Symposium 11. Lapkin, A.: Context-aware computing: A looming disruption. Research report, Gartner Inc. (24 August 2009)

[11]. Luukkala V, Niemel I. Enhancing a smart space with answer set programming. In: Proceedings of the 2010 international conference on Semantic web rules (RuleML'10). 2010 89– 103 Springer-Verlag, Berlin, Heidelberg

[12]. Marr D, Vision. A Computational Investigation into the Human Representation and Processing of Visual Informatio. W.H. Freeman and Company, New York (1982)

[13]. Mileo A, Schaub T, Merico D, Bisiani R. Knowledge-based multi-criteria optimization to support indoor positioning. In: Annals of Mathematics and Artificial Intelligence 2011: 62; 345–370.

[14]. Ramparany F. Semantic multi-sensor data processing for smart environments. In: Proceedings of the 5th International Confererence on Sensor Networks 2016 ; 199–206. Rome, Italy.

[15]. Ramparany F, H Cao Q. A semantic approach to IoT data aggregation and interpretation applied to home automation. In: 2016 International Conference on Internet of Things and Applications (IOTA). Prune, India (Jan 2016), https://hal.archives-ouvertes.fr/hal-01367543

[16]. Ramparany F, Marquez FG, Soriano J, Elsaleh T. 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). 2014;14–20. IEEE International Conference on Big Data

[17]. Ramparany F, Poortinga R, Stikic M, Schmalenstroer J, Prante T. An open Context¨ Information Management Infrastructure - the IST-Amigo Project. In: of Engineering, I.I., Technology (eds.) Proceedings of the 3rd IET International Conference on Intelligent Environments (IE'07) ; 2007 september 24-25 : 398–403. University of Ulm, Germany .

[18]. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: A practical owl-dl reasoner. Web Semant. 5(2), 51–53 (Jun 2007), http://dx.doi.org/10.1016/j.websem.2007.03.004

[19]. Sorici A, Picard G, Boissier O, Zimmerman A, Florea A.M. CONSERT: Applying semantic web technologies to context modeling in ambient intelligence. In: Computers & Electrical Engineering. No. 44 (April 2015)

[20]. Erling O, Mikhailov I, RDF Support in the Virtuoso DBMS, Springer, SCI volume 221, 2009

[21]. W3C: SWRL: A semantic web rule language combining OWL and RuleML. Available from https://www.w3.org/Submission/SWRL/ (2004)

[22]. Wagner C. Problem solving and diagnosis. In: Omega Elsevier 1993; 21: 645–656.