International Journal of Scientific Research and Management (IJSRM)
||Volume||07||Issue||02||Pages||EC-2019-189-195||2019||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47, (2016):93.67, DOI: 10.18535/ijsrm/v7i2.ec01

Rachel: An IoT smart plant based on FIWARE

Yolanda Raquel Baca Gómez[1], Hugo Estrada Esquivel[1], Alicia Martínez Rebollar[2], Daniel Villanueva Vásquez[1,3]

[1]National Council of Science and Technology,
1582, Insurgentes Sur Avenue, Benito Juárez 03940, México
[2]National Center for Research and Technological Development,
Interior Internado Palmira, Cuernavaca 62490, México
[3]Center of Research and Innovation in Information and Communication Technologies,
37, San Fernando Avenue, Tlalpan 14050, México

Abstract:
The Internet of Things (IoT) allows objects to become producers and users of information generated by themselves, by people or by other systems and also facilitates automating applications. One of the IoT topics is home automation, which allows monitoring and manipulating objects in a house. Because of the growing of IoT, different platforms are emerging to face this technology, such as FIWARE, which provides components and data models that facilitate the development of IoT applications. In this paper, a smart plant prototype based on FIWARE is presented. The plant's environment data is gathered through different sensors, and is sent to a FIWARE component called Orion Context Broker (OCB) in a FIWARE data model structure. The OCB allows to manage and publish the data, so that, the data is available in the cloud ready to be consumed by other users or applications and to automate applications.

Keywords: IoT, FIWARE, data model, Cloudino

1. Introduction

The Internet of Things (IoT) refers to a broad vision whereby "things" such as everyday objects, places and environments are interconnected with one another via the Internet [1]. In this sense, IoT is a collection of things embedded with electronics, software, sensors, and actuators, to collect and exchange data with each other. The IoT devices are equipped with sensors and processing power, which enable them to be deployed in many environments [2]. Additionally, it is expected that billions of sensors, actuators and everyday objects could be connected to the Internet, thus being able to report about the conditions in their surroundings and act on their environments. These interactions will mostly happen without human intervention, paving the way for smart objects and applications that are able to measure, regulate and optimize their environments or their own operation [3].

IoT provides connectivity for anyone and anything wherever and whenever. With the advancement in technology, we are moving towards a society, where everything and everyone will be connected [4]. One of the many applications of IoT is the home automation, which allows to control and to monitor lights, ventilator, gas leakage, motion detection, watering garden and so on from a smartphone anywhere. Therefore, an automated and quantified system to take care of plants could be a very useful tool [5].

Due to the growing awareness of IoT, IoT platforms have been raised as well, such as FIWARE[1] which is an emerging IoT platform, funded by the European Commission (EU), which is pushing for an ecosystem providing APIs and open source implementations for lightweight and simple means to gather, publish, query and subscribe context-based, real-time "things" information [6]. More than 100 European cities are already using FIWARE, most of them for IoT and mobility solutions. Moreover, one of the main advantages of FIWARE is that they are programming-based approaches, where solutions are generated starting from low design levels [7].

In this paper, an IoT smart plant prototype based on FIWARE is presented. The plant can sense its own environment through some sensors. The generated data is collected and processed by using Cloudino and Arduino devices. Then, the data is sent to the Orion Context Broker by using the AirQualityObserved FIWARE data model. Finally,

the data can be used to automate some functions in the environment's plant and the data is available to be consumed by other applications. This smart plant prototype involves the entire cycle of the data, from the development an IoT device to gather and process the data until its publication in the Orion Context Broker using a data model, where the data can be consumed for different purposes. Finally, an application to visualize the data is built.

2. Rachel IoT smart plant architecture

This paper presents a prototype of an IoT smart plant based on FIWARE technologies, Cloudino and Arduino devices to publish the data and automate different actions. In Figure 1, the Rachel IoT architecture is presented; which shows the interaction between the components of the prototype.

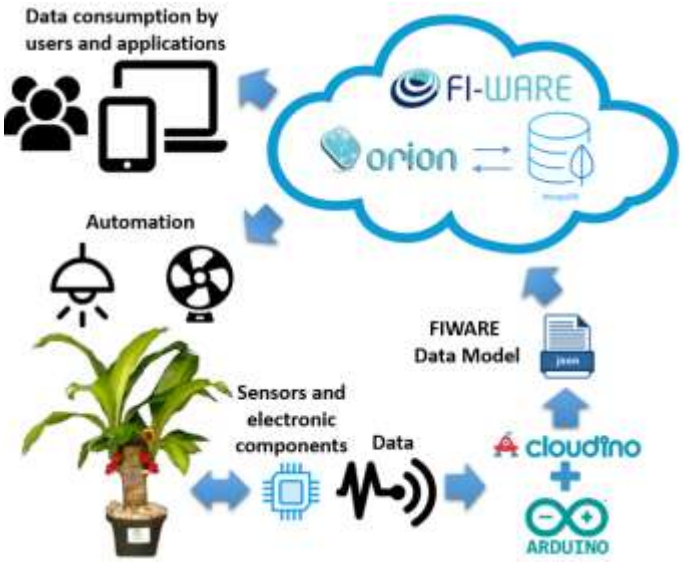

Figure 1: Rachel IoT smart plant architecture

The prototype consists of a plant called Rachel, with some sensors on it; the sensors obtain data from the environment; then, the gathered data is processed by using Cloudino and Arduino devices and it is transformed into a FIWARE data model. Finally, the transformed data is sent to the Orion Context Broker. Once the data is available in the Orion Context Broker, it can be used to execute some actions or it can be consumed by other users and applications.

3. Methodology

In this section, the followed methodology for the development of the Rachel IoT prototype is described. The steps are shown in Figure 2, each is briefly described in the following subsections.

Figure 2: Rachel IoT development methodology

3.1 Connection of the sensors and electronic components

The sensors connected to obtain the data of the plant's environment and the electronic components used in the prototype are listed below. In Figure 3 a photograph of the prototype is shown.

- **Soil Moisture sensor FC28.** This component is used to measure the humidity of the ground where the plant is seeded. The sensor provides values between 0 and 940, where 0 represents total dryness and 940 represents the highest value of humidity.
- **Temperature sensor DHT11.** This component is used to measure the temperature of the environment in Celsius degrees.
- **Light sensor photoresistor.** This component is used to identify if there is light or not in the environment. The sensor provides two values 1 and 0, where 1 represents light and 0 represents no light.
- **Infrared sensor.** This component is used to identify if there is an object in front of the plant. The sensor provides two values 1 and 0, where 1 represents presence and 0 represents no presence.
- **Relative Humidity Sensor DHT11.** This component is used to measure the humidity of the environment. This value is sent to the Orion Context Broker and it is only shown in the LCD Display.
- **8x8 LED matrix with MAX7219.** This component is used to simulate emotions of the plant.
- **4 Channel 5V Relay Module**. This component is used to turn the electrical components on and off.
- **Display LCD 2x16.** This component is used to show the values of the sensors. Therefore, any person who is nearby Rachel could see the values of the sensors.

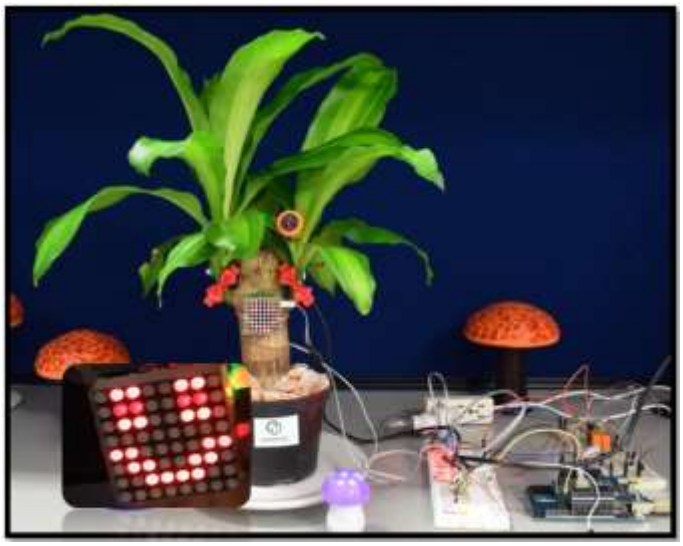

Figure 3: Rachel IoT prototype

3.2 Definition of the data model and creation of the entity in the Orion Context Broker

FIWARE provides data models[2] in order to facilitate the exchange of data between applications. These data models have been defined in relation to the FIWARE reference context model: Open Mobile Alliance Next Generation Service Interfaces (OMA-NGSI).

The standard NGSI v2 of FIWARE is intended to manage the entire lifecycle of context information, including updates, queries, registrations, and subscriptions. The main elements in the NGSI data model are context entities, attributes and metadata, as shown in Figure 2. The difference among these elements is that attributes describe the entity and metadata describe attributes [8], [9]:

Figure 4: Elements of the NGSI data model

Context entities, or simply entities, are the center of gravity in the FIWARE NGSI information model. An entity represents a thing, any physical or logical object (e.g., a sensor, a person, a room, an issue in a ticketing system, and so on). Each entity has an entity id and entity type. Entity types are intended to describe the type of thing represented by the entity. Each entity is uniquely identified by the combination of its id and type. Context attributes are properties of context entities. In the NGSI data model, attributes have an attribute name, an attribute type, an attribute value and metadata.

In the prototype data model design, an entity based in the *AirQualityObserved*[3] FIWARE data model and in the NGSI Specification has been created, with the purpose to send the Rachel information to the Orion Context Broker. In Figure 5, the Rachel entity data model is presented.

```
{
 "id": "Rachel",
 "type": "AirQualityObserved",
 "address": {
  "addressCountry": "MX",
  "addressLocality": "Ciudad de México",
  "streetAddress": "Av. San Fernando 37"
 },
 "dateObserved": "2016-03-15T11:00:00",
 "location": {
  "type": "Point",
  "coordinates": [19.2914188,-99.165384]
 },
 "source": "Rachel IoT Application",
 "relativeHumidity": 0.54,
 "temperature": 12.2,
 "soilMoisture": 900,
 "Light": 1,
 "Presence": 1
}
```

Figure 5: Rachel IoT data model

The "id" field uniquely identifies the entity. The "type" field establishes the data model, in this case the *AirQualityObserved* data model. The "address" field specifies the address in which Rachel is located. The "dateObserved" field shows the date and hour when the data is sent to the Orion Context Broker. The "location" field specifies the coordinates where Rachel is located. The "source" field establishes the data source; in this case, the data is obtained from the Racel IoT Application. And, finally the "relativeHumidity", "temperature", "soilMoisture", "Light" and "Presence" fields show the values of the sensors measurements.

3.3 Data processing with Cloudino and Arduino

The data is processed by using Cloudino and Arduino. The necessary code for the functioning of the prototype can be loaded in the Arduino through the Cloudino's interface in a remote way. Additionally, Cloudino offers different functions that facilitate the programming of Arduino and it also provides WiFi connection to the prototype. So that, the prototype is capable of constantly sending the data to the Orion Context Broker. A guide called "Creating your Air Quality Sensor with Cloudino[4], which is published as a part of the SmartSDK project, has been taken as a reference for the

[2] https://github.com/Fiware/dataModels/blob/master/specs/howto.md

[3] https://fiware-datamodels.readthedocs.io/en/latest/Environment/AirQualityObserved/doc/spec/index.html

development of the prototype. There are sections in the code for different settings, such as set the libraries required for each sensor, pines configuration, variables, functions definition, and so on. In Figure 6, an example of the code in the Cloudino´s interface is presented. The function presented is used to obtain the data from the sensors.

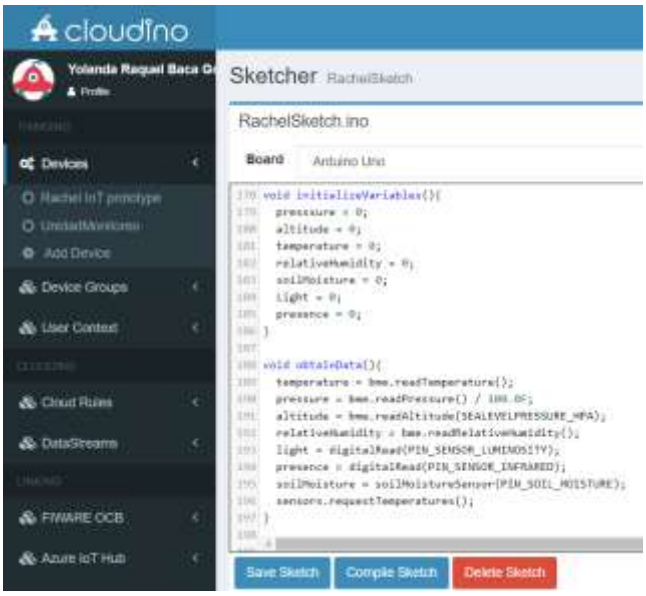

Figure 6: Example of the code in the Cloudino's interface

3.4 Data stored in the Orion Context Broker

The gathered data from the sensors through the Cloudino and Arduino devices are sent to the Orion Context Broker.

The Orion Context Broker is an implementation of the NGSI REST API and is the key component of FIWARE to enable the data context ingestion and also to enable the subscription of applications to the data context. The main concept of the Context Broker is that data context producer can generate information and place this in the cloud, without a previous knowledge of the users or application that will use the data. The Orion Context Broker is able to handle context information in a large scale by implementing standard REST APIs and allows developers to manage the whole lifecycle of context information including updates, queries, registrations and subscriptions [8], [10], [11], [12].

There is a function in the Cloudino's interface that allows to transform the data in the format allowed by the Orion Context Broker. In Figure 7, the example of the code for the data transformation in the *AirQualityObserved* data model is shown.

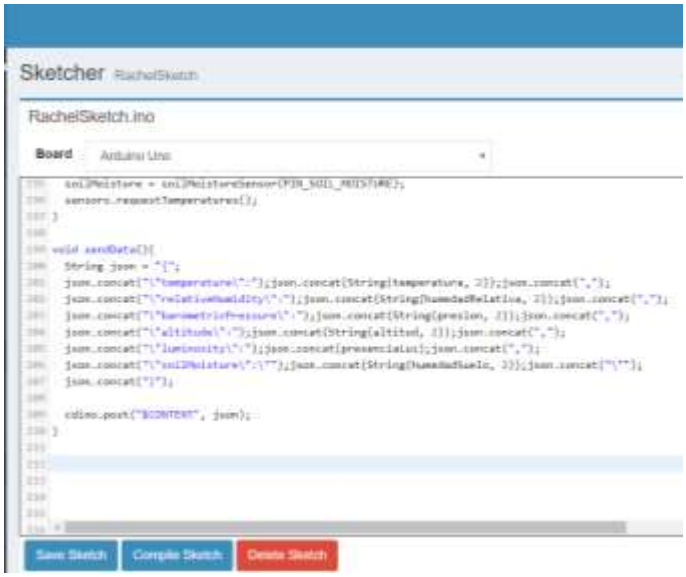

Figure 7: Function to transform the data in the format allowed by the Orion Context Broker

The Cloudino's interface provides a function to automatically send the data to the Orion Context Broker. There is a small form to be filled out in order to start sending the gathered data by the sensors to the Orion Context Broker. In Figure 8 an example of the form is shown. It is necessary to specify the name of the device, in this case Rachel IoT prototype, the entity ID, the entity definition, the URL of the Orion Context Broker. The Auth URL, Auth User and Auth Password if needed, the Orion Context Broker can be configured to require authorization.

Figure 8: Function to send the data to the Orion Context Broker

3.5 Prototype automation

The generated data by the Rachel IoT prototype is available in the Orion Context Broker and can be used to automate some functions, such as:

- **Plant emotions.** The LED Matrix is used to show, as default, a happy face in the electronic display. Then, a sad face is shown when the soil moisture sensor has a value that represents dry ground, and a heart beating is shown when the infrared sensor has a value that represents the presence of an object, in this case, we assume that a person is getting close to Rachel. With this component it is simulated that the plant can express some emotions according to the changes in its environment.
- **Automatic irrigation system.** If the soil moisture sensor has a value that represents dry ground, a sad face is shown in the LED Matrix. There are two possible scenarios: in the first, the irrigation system turns on until the value of the sensor represents wet ground; in the second, the irrigation system stays off in order to bring the user the opportunity to interact with Rachel and give her water. In both cases, when the sensor represents wet ground, the happy face appears in the LED Matrix.
- **Turning on and turning off a ventilator.** When the value of temperature exceeds 25°C the ventilator turns on, simulating that Rachel is hot, and when the temperature is under 25°C the ventilator turns off and remains off until the temperature raises.
- **Turning on and turning off a lamp.** If there is no light the lamp turns on, simulating that Rachel needs light to grow. If there is light, the lamp turns off and remains off until there is no light.
- **Reacting when someone is close.** When an object is detected in front of Rachel, it is assumed that it is a person getting close to Rachel, and the LED Matrix shows a heart beating. When the person goes, a happy face appears on the LED Matrix, and the happy face remains until another event happens.

3.6 Data consumption by users and applications

The stored data in the Orion Context Broker can be consumed through subscriptions. A subscription is a POST method that allows applications getting asynchronous notifications. This way, it is not necessary to continuously repeat query requests; the Orion Context Broker will send information when it comes. Some data need to be specified in the subscription: (1) the entity from we want to obtain the information, (2) the data in the entity that must change to trigger the subscription, (3) the data that is going to be sent to an application or service, and (4) the application or service which is going to receive the data. In this case, a subscription that allows sending the data from the Rachel entity to the Rachel IoT Application has been created. The elements of the Subscription to the Rachel entity are shown in Figure 9.

```
{
 "description": "Rachel IoT Prototype",
 "subject": {
 "entities": [
 {
  "id": "Rachel",
  "type": "AirQualityObserved"
 }
 ],
 "condition": {
 "attrs": [ "relativeHumidity",
            "temperature",
            "soilMoisture",
            "Light",
            "Presence"]
 }
 },
 "notification": {
  "http": {
  "url":
"http://www.greenroute.com:8080/notifications",
  "method": "POST"
  },
  "attrsFormat": "keyValues",
  "attrs": [ "id"
            "relativeHumidity",
            "temperature",
            "soilMoisture",
            "Light",
            "Presence"]
 },
 "expires": "2020-04-05T14:00:00.00Z",
 "throttling": 1
}
```

Figure 9: Subscription to the Rachel entity

When the values of the sensor measurement change in the Rachel entity registered in the Orion Context Broker, the Rachel IoT Application will get an asynchronous notification. This way, it is not necessary to continuously repeat query requests.

The "entities" field specifies the entity from which we need to receive notifications. The "condition" field contains the attributes that must change to send the notification. The "notification" field specifies where the notification will be sent. The "attrs" field specifies the values that will be sent, could be one or more attributes of the entity. The "expires" field specifies the date when the subscription will stop working. Finally, the "throttling" field specifies the minimal period of time in seconds which must elapse between two consecutive notifications.

4. Results and Discussion

The Internet of Things is an integral part of the Internet of the Future along with other technological trends like Big Data, Cloud Computing, Mobile Computing and Augmented Reality. In addition, these types of projects allow bringing to life to objects and allow them to interact with their environment in an autonomous way, without the need of human help. The Rachel IoT application allows to process information and modify its environment, and to send data to the cloud in an autonomously way by using sensors and the FIWARE platform.

The FIWARE platform, brings tools to avoid the constant execution of requests to obtain the changes on the sensors measures. As soon as a value changes, automatically, through the subscription to the Orion Context Broker, Rachel sends the values to the endpoint designed to receive the notifications. In this case, the values are sent to the Rachel IoT Application. In the application the values of the sensors are shown. The light bulb icon shows the current value: with yellow *light on* and with gray *light of*. By clicking on the light bulb icon it is possible to turn on or turn off the light. With the start button it is possible to start the irrigation system. In Figure 10 the Rachel IoT application is shown.

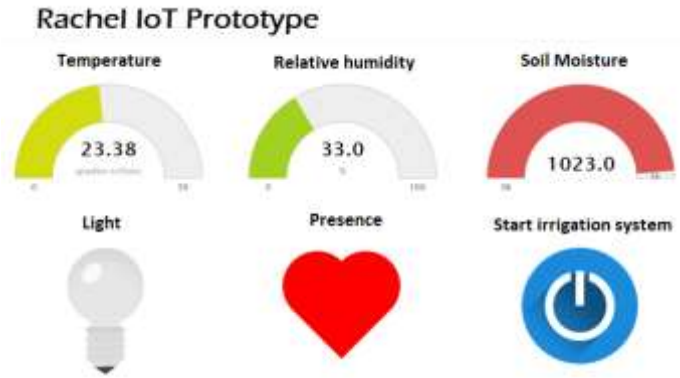

Figure 10: Rachel IoT application

However, it is possible to build any kind of applications in order to use or show the data. Moreover, a generic data model is used, thus, it facilitates to share and publish the data. Anyone can use the published data in the Orion Context Broker, and use it in their own applications.

5. Conclusions

The FIWARE platform offers data models and services that simplify the handling of the data. Therefore, it is possible to develop different kinds of intelligent applications by using FIWARE, and, thus share information in the cloud and automate processes. In addition, using the FIWARE data models facilitate the consumption of data by any application and ensure interoperability between different applications. The Rachel IoT smart plant is an example of how the information from several sensors can be shared by using FIWARE technologies, and how the information can be used in a specific application.

References

[1] T. L. Koreshoff, T. Robertson y T. Wah Leong, «Internet of Things: a review of literature and products» de Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, 2013.

[2] Y. Yang, L. Wu, G. Yin, L. Li y H. Zhao, «A Survey on Security and Privacy Issues in Internet-of-Things» IEEE Internet of Things Journal, vol. 4, nº 5, pp. 1250-1258, 2017.

[3] J. Gubbi, R. Buyya, S. Marusic y M. Palaniswami, «Internet of Things (IoT): A vision, architectural elements, and future directions» Future generation computer systems, vol. 29, nº 7, pp. 1645-1660, 2013.

[4] R. Khan, S. Ullah Khan, R. Zaheer y S. Khan, «10th International Conference on Frontiers of Information Technology» 2012.

[5] B. Sandhya, M. Pallavi y M. Chandrashekar, «IoT Based Smart Home Garden Watering System Using Raspberry Pi 3» International Journal of Innovative Research in Science, Engineering and Technology, vol. 6, nº 12, pp. 101-106, 2017.

[6] A. Ahmad, F. Bouquet, E. Fourneret, F. Le Gall y B. Legeard, «Model-Based Testing as a Service for IoT Platforms» de 7th International symposium on leveraging applications of formal methods, verification and validation, 2016.

[7] H. Estrada, K. Nájera, B. Vázquez, A. Martínez, J. C. Téllez y J. J. Hierro, «Applying Tropos modeling for Smart mobility applications based on the FIWARE platform» de Proceedings of the Ninth International i* Workshop, 2016.

[8] A. Martínez, F. Ramírez, H. Estrada y L. A. Torres, «Generic module for collecting data in Smart Cities» de 2nd International Conference on Smart Data and Smart Cities, Puebla, 2017.

[9] FIWARE, «FIWARE-NGSI v2 Specification» 2018. Online]. Available: http://fiware.github.io/specifications/ngsiv2/stable/. [Accessed: Feb. 20, 2018].

[10] FIWARE, «FIWARE.OpenSpecification.Data.ContextBroker» [Online]. Available: https://forge.fiware.org/plugins/mediawiki/wiki/

fiware/index.php/FIWARE.OpenSpecification.Data.ContextBroker. [Accessed: Feb. 20, 2018].
[11] FIWARE, «Publish/Subscribe Context Broker - Orion Context Broker» 2018. [Online]. Available: https://catalogue-server.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker. [Accessed: Feb. 20, 2018].
[12] FIWARE, «Welcome to Orion Context Broker» [Online]. Available: https://fiware-orion.readthedocs.io/en/master/. [Accessed: Feb. 20, 2018]