An open and lightweight Internet of Things platform to facilitate the development of context-aware applications.

FERNANDO RIBEIRO[1,2], JOSÉ METRÔLHO[1]
Informatics Department
[1]Instituto Politécnico de Castelo Branco
[2]ALGORITMI Research Centre, University of Minho
PORTUGAL
fribeiro@ipcb.pt, metrolho@ipcb.pt

Abstract: - The Internet of Things (IoT) has encouraged the development of innovative applications and services that make use of the enormous amount and diversity of data provided by such things to provide novel and useful services. However, to be successful, these platforms have to meet the expectations of different stakeholders, namely application developers and end-users. In this work we analyze some previous works with similar goals and we discuss some issues that need to be addressed when developing an IoT platform. Based on this knowledge, we proposed and developed an open and lightweight IoT platform that allows application developers and end-users to integrate IoT objects, to create new sensor data models monitoring and managing objects and that allows developers to access and use, in their applications, high level APIs to handle the information related to the IoT objects. In this approach there is particular focus on two characteristics: simplicity and flexibility. Simplicity, allowing application developers to easily integrate their own objects using their own semantics, and allowing end-users to visualize and manage their objects through the web portal. Flexibility, due its open nature that supports the integration of new types of objects and data formats, it allows developers to customize their own devices and models. The first experimental results indicate that this is a viable approach to manage Things on the Internet and to simplify the development of context-aware applications.

Key-Words: ambient intelligence, context-awareness, internet of things, open platform, smart spaces, ubiquitous computing, web of things.

1 Introduction

The Internet of Things (IoT) is defined as a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on existing and evolving interoperable information and communication technologies [1, 2]. Or, in other words, the IoT could allow people and things to be connected at anytime, anyplace, with anything and anyone [3]. The IoT makes full use of "things" to offer services to all kinds of applications [1, 2]. These “things” are objects of the physical world or the information world, which are capable of being identified. When embedded in smart environments they can provide support for people in their everyday life activities. Enabling easy access and interaction with a wide variety of things (e.g. actuators, displays, cameras, wearables, smart devices, mobile devices, and a huge diversity of sensors, with computational and communicational capabilities) and between things themselves, the IoT will encourage the development of a number of innovative applications and services, that make use of the large amount and diversity of data provided by such things to provide novel and useful services. However, to succeed it is important to facilitate the development of context-aware applications and it is necessary to develop interfaces between the physical world and end users, gaining new knowledge about the environment, people and their activities and thus set up innovative and useful services and applications. In this context, the challenges to develop smart and context aware applications for the IoT are immense. Discovering and connecting to smart objects and external information resources, developing interfaces to connect with the heterogeneity of real world smart objects and other issues related to security and privacy, are some of the challenges. IoT platforms can play an important role, interconnecting devices onto the Internet as web resources, using traditional Web standards, managing them, facilitating access to the objects’ capabilities and facilitating the design and implementation of context aware applications, allowing developers to easily access and use information available on the IoT and providing them with a set of tools to enrich their knowledge about the context and thus making their applications smarter and more useful.

In this paper we present an open and lightweight IoT platform that facilitates the integration and management of IoT objects, making information about them available to be used by context-aware applications. With this platform it is not intended to provide a complete middleware solution for managing "things" on the Internet. Instead, the goal is to provide support for context-aware application developers to easily integrate, manage and use information about standard and user customized objects. Application developers are able to access existing objects and also to integrate their own objects using standard or customized protocols. The platform processes the information about objects it stores it, and provides tools to manage them and to make them available to be used by external applications.

Results from initial experiments suggest that the proposed IoT platform can be used to easily associate and integrate Things on the Internet, to manage them and to make them available to be used by context aware applications developers.

The remainder of this paper will be as follows: Section 2 presents a brief review of related work; Section 3 we present the motivation for an open and lightweight IoT platform and the main challenges involved; Section 4 provides an overview of the general guidelines related to IoT platforms; Section 5 we describe our proposal, a lightweight Internet of Things platform to facilitate the development of context-aware applications; Section 6 presents some functionalities of the platform;Section 7 we describe some examples of context-aware applications to illustrate the usefulness of the platform and its viability, making the development of context aware applications easier; and finally, in Section 8, we present some conclusions and we outline some of the future work.

2 Related work

Several studies have addressed the development of new middleware and platforms for the IoT. In this section we provide a brief overview of some works that, in one sense or another, integrate and manage objects in the IoT and make them available to be used by external consumers, allowing developers to focus on the development of the context aware application and release them from the complexity of sensor deployment.

The CUBIQ[4] is a common and shared platform which mediates diversified sensor systems and ubiquitous applications. It aims to make ubiquitous application development easier, drive context-data distribution and integrate various types of previous independent ubiquitous systems as an infrastructure. The CUBIQ architecture consists of three layers: 1) a data access layer that is responsible for transparency for representation, location, mobility, migration, replication and persistency of context data defined in many types of sensor systems; 2) the intra-context processing layer, deals with data dissemination and processing with real-time and scalability and 3) the inter-context processing layer, that is responsible for service federations. This platform enables devices and services to connect to each other to provide context-aware services, and proposes an architecture to federate devices by establishing features such as a RESTful interface and a P2P-based resource discovery mechanisms. The Collaborative Open Market to Place Objects at your Service (COMPOSE) [5, 6] is a framework that uses cloud computing infrastructures and IoT technologies allowing for the integration of smart objects and external services as well as the provision of scalable resources for data and application management. It is an open marketplace for developing, deploying and maintaining IoT applications in a secure, cloud-based platform. It enables the end-to-end development and deployment of context-awareness by providing a set of tools and methods for the collection of contextual information on smart devices, tools for communication with external resources, infrastructure for hosting the data storage and processing, and an open marketplace infrastructure, in which smart objects are associated to services that can be combined, managed and integrated to build innovative applications. ThingSpeak [7] is an open data platform and API for the Internet of Things that enables you to collect, store, analyze, visualize, and act on data from sensors or actuators, such as Arduino, Raspberry Pi, BeagleBone Black and other hardware. It can trigger several types of responses and actions on remote individual hardware, storing numeric and alphanumeric data, numeric data processing, location tracking, and status updates. It supports real-time data collection and storage, MATLAB analytics and visualizations. Carriots [8] is a Platform as a Service (PaaS) designed for Internet of Things (IoT) and Machine to Machine projects. This platform makes it possible to connect devices (e.g. Arduino, Raspberry Pi, Nanode), collect data from connected objects, store it, and build applications. It provides an API hosting and a development environment to support application development through the HTTP RESTful API to push and pull XML or JSON data and it provides support to deploy, interact, enable or disable devices.CA4IOT [9] is a middleware that is conceived to help users by automating the task of selecting the sensors according to the problems/tasks at hand. Its focus is on automated configuration of filtering, fusion and reasoning mechanisms that can be applied to the collected sensor data streams using selected sensors. The CA4IOT architecture consists of four layers: data, semantics and context dissemination layer, processing and reasoning layer, context and semantic discovery layer and sensor data acquisition layer. The FP7-287305 OpenIoT project [10] provides a middleware platform enabling the semantic unification of diverse IoT applications in the cloud. It uses the W3C Semantic Sensor Networks ontology, which provides a common standards-based model for representing physical and virtual sensors. It includes a sensor middleware to support the collection of data from sensors and the proper semantic annotation. Furthermore, it provides an integrated environment for building/deploying and managing IoT applications and it offers a set of visual tools that enable the development and deployment of IoT applications, enabling: visual definition of IoT services; visual discovery of sensors according to their location and type; configuration of sensor metadata; monitoring of the status of the various IoT services; visualization of IoT services on the basis of Web2.0 mashup. The openIoT [11] is an open service framework for the IoT which is proposed to facilitate entrance into the IoT-related mass market, and establishing a global IoT ecosystem with the worldwide use of IoT devices and software. It consists of three server side platforms. Planet Platform, supports IoT device's registration, management and discovery. When an IoT device is registered with Planet Platform, device profiles including the device's ID, name, keyword, location, and network address are delivered to Planet Platform. Mashup Platform plays a role as data repository and also provides mash-up service. The data in Mashup Platform can be accessed through open APIs. The Store Platform contains applications or links to Web address that provide user services through interaction between IoT devices or Mashup Platforms. There is also one device side platform named Device Platform to help connecting and cooperating things to Open IoT platforms and application software. All platforms have open APIs based on RESTful interface to allow developers to easily make their own software using the APIs. MAGIC Broker 2 [12] is a platform designed to offer a web based API for building IoT applications. It supports publish-subscribe event channels, persistent content and state storage, and brokerage of services via remote-procedure call. It includes a sensor/actuator network portal that allows users to contribute information about sensors or actuators that send data from a sensor to the portal. Other users can find the sensor data they are interested in to create new applications and dashboards using the portal's processing and visualization features. This platform can also broker synchronous two-way request-response interactions called services with devices registered with the platform.

We identified several IoT platforms and middleware solutions that support integration of objects into the Internet providing access of their data, manage them and make their information available to be used by external applications. Some IoT platform focuses on different aspects such as device management, interoperability, semantic modelling, context-awareness, security and privacy, etc., others try to provide a full solution trying to address all aspects. Our goal is not to provide a complete and full solution. Instead, our goal is to design an IoT platform to facilitate the development of context-aware applications. We adopt an open and lightweight approach allowing application developers and end-users to integrate, to manage and to use information about standard and user customized objects, using their own semantics.

3 Challenges and motivation for an open and lightweight IoT platform

Deploying context-aware information systems for the IoT poses many challenges that are not usually found when developing traditional information systems and that may limit their applicability and success. The singularities and specificities of these environments make them very rich and heterogeneous in terms of devices (sensors and actuators), interaction mechanisms and they create new opportunities for several innovative applications. However, these also simultaneously create additional challenges. When large numbers of sensors and actuators are required, the traditional approaches (i.e. handle with sensors and actuators directly with applications individually and manually) become infeasible. In these scenarios, IoT platforms may represent an important role and they can be used and configured for multiple and different applications.

3.1 Challenges

In this section, we consider some of the major research challenges when developing IoT platforms to support the development of context-aware applications. In addition to the hardware and technical challenges related to communication and networking, latency, power and storage, there are a number of other challenges that may influence the development of platforms to support the development of context-aware applications, and they should be considered:

- Security and access control: platforms should provide mechanisms to ensure that only authorized users, devices and applications can access the platform and its data. IoT enables a constant transfer and sharing of data between Things and users in order to achieve particular goals. In these environments, authentication, authorization and access control are important to ensure secure communication and information manipulation [13-16].
- Trust and privacy: in the domain of trust, some concerns are related to the surveillance of users without knowledge or consent, the exchange of sensitive data, the control of access and ownership of data and sensitive context [15-17]. IoT introduces new ways of collecting and processing information from objects and from different sources which very often reveals information about the individuals, namely information related to their location, activities, habits or interactions with others. For all of these situations, platforms should provide users with mechanisms to control what personal data may be collected or accessed and who may collect and access their personal information.
- Processing and mining large volumes of data: platforms should be able to process and mine large volumes of data from many different sources to provide useful services [18].
- Managing heterogeneity: managing the high diversity of technologies, devices, environments and applications is a significant challenge [18].
- Discover and integrate sensors and services: there is the need for efficient mechanisms to support sensor discovering, and integrating them into the platform. Thus, platforms should provide mechanisms to easily and efficiently integrate objects.
- Publish/subscribe: software platforms should provide mechanisms to easily support publishing/subscribing services facilitating the use of data by external applications.
- Facilitate software development: the availability of efficient tools to be used by external application developers is an important issue. Designing and building context aware applications for IoT will require enormous effort and device particularities, networking and embedded software knowledge. Thus, tools to manage devices and services configure permissions and access historical data provided by IoT platforms are important resources and may contribute to facilitate software developers to build their applications quickly and with higher quality.

3.2 Motivation

Objectively, to be successful platforms these must support the development of context aware applications for the IoT and they should support many types of information, distinct data formats provided from many different sensors, and should also be adaptable to the real characteristics and needs of each specific context and developer. They should be provided with appropriate tools and mechanisms to facilitate the task of developing context aware applications, including support to easily integrate new sensors and different types of data and appropriate API to control and access objects as well as to obtain contextual information in the appropriate way. Thus, in our approach we decide to specifically focus on two characteristics: simplicity and flexibility. To achieve simplicity our proposal does not use semantic modelling or device models. Instead, we adopt an open and lightweight approach allowing application developers to easily integrate their own objects using their own semantics, and allowing end-users to visualize and manage their objects through the web portal. To address flexibility, our IoT platform is not limited to standard devices or user models, instead it is open to new types of objects and data formats, allowing developers to customize their own devices and models.

4 Internet of Things platforms

IoT platforms are very common and they are usually developed to support the integration of data from several sensors and actuators into the Internet. Frequently they are focused on the communication and networking aspects between devices that are used for sensing and acting over real world objects. However, these platforms can also play an important role in helping context aware application developers to deal with the particularities of each device and with a variety of protocols to support most representative technologies, providing the mechanisms to deal with data persistence, security, privacy as well as provide a uniform and open interface to access data through well-known web technologies. In other words, they may help context-aware application developers to focus on the development of their applications and release them from dealing with the problems of context information acquisition, modeling, and management. To achieve these goals IoT platforms should provide support to easily integrate sensors and actuators allowing their owners to configure them, allowing them to define new types of information. Moreover, platforms should provide support for processing and storing information and should provide mechanisms to manage this information and to make it available to be used by external applications. Additionally, the platform should have an open architecture and provide developers with appropriate tools to allow them to access and use this information in their applications. Specifically, platforms should combine two different perspectives: *i)* information producers, *ii)* information consumers and, among others, they should be designed to:

- Facilitate connectivity among heterogeneous IoT-based networks.
- Be flexible to support new types of objects and data.
- Provide open API to manage different objects and to appropriately answer to the needs of consumer applications.
- Adequately manage data according to their private or public nature.

Fig. 1 presents a general architecture of an IoT platform.

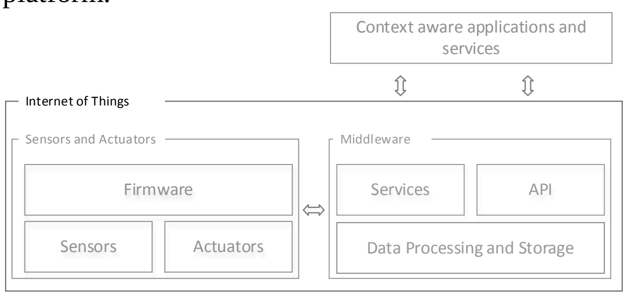


Fig. 1. General system architecture of the IoT Platform

5 The proposed open and lightweight IoT platform

The proposed open and lightweight platform will facilitate the development of context-aware applications, enabling:

- The integration of IoT objects using standard protocols
- Application developers to create new sensor data models
- Manage objects and provide mechanisms to ensure that only authorized users, devices and applications can access the platform and its data
- Store large volumes of data from objects and process and mine large volumes of data from many different sources to provide useful services
- End-users and application developers monitoring and managing objects through the IoT web portal
- The easy access and usage of high level APIs to handle the information related to the IoT objects.

Its architecture is depicted in Fig. 2. The five main components of the platform are: sensors and actuators, data processing and storage, API, web portal and context-aware applications.

Sensors and actuators: This module includes sensors, actuators and software that is responsible for controlling, monitoring and manipulating the data provided by the objects, and then sends this data to the data processing and storage module. Each object should be able to communicate its own data to the data processing and storage module. Objects are registered in the middleware by the owner through the web portal or, autonomously, including in their message to the middleware the owner ID, location, type of data and correspondent value. It is also possible to define the level of access to the data of each object. In this case the message should include a specific field to indicate if the data is private or public. This information is sent to the Data Processing and Storage module using specific formats.

At this stage the platform is able to support Arduino and Raspberry pi based objects (e.g. RFID, Bluetooth, Temperature, Humidity, displays). The platform is able to accept objects according to owner specifications, i.e. the owner may define their own type of data models, defining the data type, data format and access level.

Data processing and storage: This module receives the information provided by the objects. It acts as a context repository and it stores information in a way that queries so information can be efficiently handled when needed. It is also responsible for processing and storing the information and for controlling the access to this information through the API.

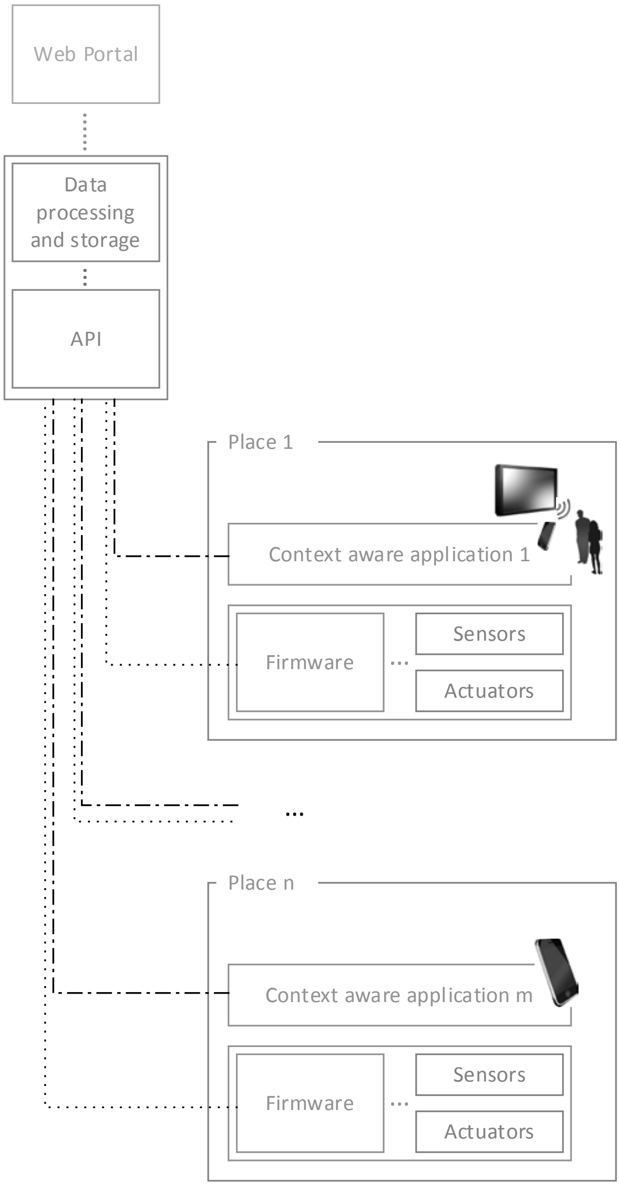


Fig. 2. System architecture.

API: This module includes a set of functions that can be accessed by external applications and services to get information about objects on the Internet. It includes a set of functions to access information about objects, locations, and about historical data related to each object. This module also provides a set of functions to control some actuators.

Web Portal: It provides the interface to access the data of the objects to manage them and control the accessibility to their data. This module, besides acting as a portal to visualize the information of each sensor or location, it also allows sensor configuration and provides support to manage access to the information from each sensor or location, namely the access to the API functions.

Context-aware applications: Application developers are able to access the API functions according to their permissions and to integrate them into their applications. Through the API they are able to connect to external objects and to access contextual information according to the specified object or location. Additionally, they are able to ask the platform about historical data and this information may be used to discover patterns or habits. At this stage the set of functions is not fully developed. We need to expand it to support statistically elaborated functions over the historical data

6 Experimental work

The experimental work comprised three experiments to assess its overall operation. The goal was to validate the concepts theoretically described, mainly aspects related to the working issues of the core of this system, namely:

- How easy is it to integrate different Things in the platform.
- How it helps in monitoring and configuring Things.
- How easy is it to consume information from the IoT platform in the development of new context aware applications.

6.1 Using the IoT platform to integrate Things

As referred before, Things can be integrated in the platform using two different methods. One method is for the owner to register their Things through the Web Portal. To do this, the owner needs to fill in the form with the object information: location, type of data and the time period for sending readings. Before configuring the Thing, the user needs to select if they intend to add a sensor or an actuator. Figure 3 presents the form to add a new sensor. After this operation, the system provides a ThingID to be used by the owner.

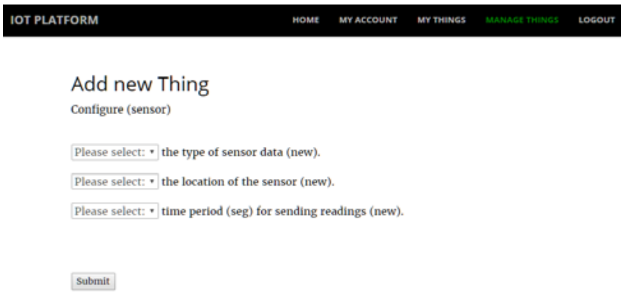
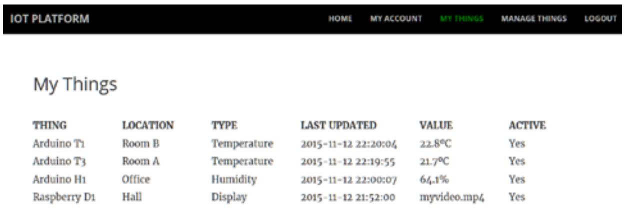


Fig. 3. Using the web portal to add new sensor.

Alternatively, the Thing may register itself in the platform. In this case, Things should send the data to identify the correspondent owner, location, type of data and level of access to the platform. Arduino based sensor (e.g. Arduino Yún and DHT22 digital temperature and humidity sensor DHT22) runs the standard code for reading DHT22 and the user adjusts the message fields to correctly identify the sensor type, location and access level. By utilizing this method users are able to define and use customized objects.

6.2 Monitoring and configuring Things through the IoT web portal

The Web Portal provides the user with an interface to configure and monitor their things. Fig. 4, shows the list of all Things belonging to the user, but the web portal also allows users to personalize their queries about objects, namely selecting the type of Things or location.



THING	LOCATION	TYPE	LAST UPDATED	VALUE	ACTIVE
Arduino T1	Room B	Temperature	2015-11-12 22:20:04	22.8ºC	Yes
Arduino T3	Room A	Temperature	2015-11-12 22:19:55	21.7ºC	Yes
Arduino H1	Office	Humidity	2015-11-12 22:00:07	64.1%	Yes
Raspberry D1	Hall	Display	2015-11-12 21:52:00	myvideo.mp4	Yes

Fig. 4. List of *MyThings*.

Through this portal, users are able to add or remove Things from their accounts, configure parameters of their things (e.g. time period between readings for humidity sensors) and visualize data and historical data about each Thing. Fig. 5 shows how to select the Thing to visualize.

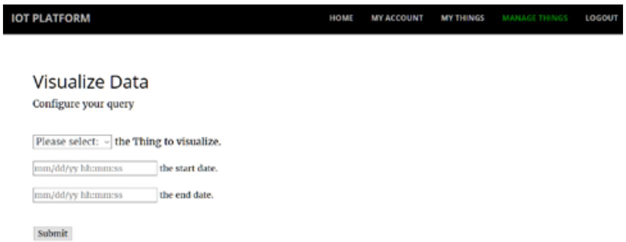


Fig. 5. Select the Thing to visualize.

Additionally, it is possible to analyze historical data related to each Thing and personalize the time period that the user is intending to visualize. This can be shown in a table or in a chart. Fig. 6 presents the historical data between a customized start date and end date related to a humidity sensor.

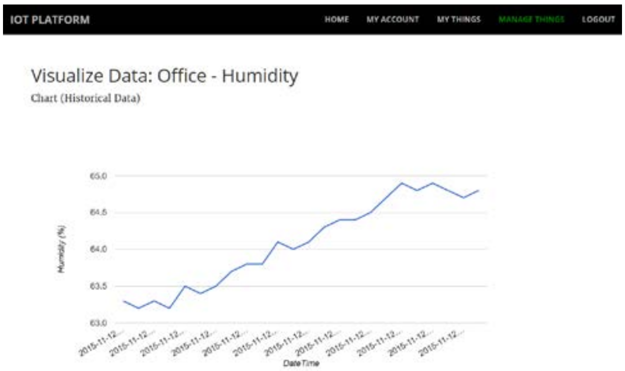


Fig. 6. Graph with historical data related to a humidity sensor.

6.3 Accessing things using the API provided by the IoT platform

Information about Things can be used as contextual information associated to the environment where they are situated and it can be handled through the platform and thus accessed by context-aware application developers. The API, provided by the platform, allows context-aware application developers to get the information they want to use in their applications. Using these functions, it is possible to obtain information about a specific location and/or types of information and about some historical and statistical data. Below are some examples of functions to obtain information about Things:

– Get information about the user's Things.
  Return: List of Things (ThingID, type of Thing, location). Empty if none.

– Get information about all Things in the location ID "Classroom".

Return: List of Things (ThingID, type of Thing). Empty if none.

– Get information about "Temperature" on the location ID "Classroom".
Return: List of Temperatures, if more than one Temperature Thing in this location. Empty if none.

– Get information about all Bluetooth Things in the location ID "Classroom".
Return: List of Bluetooth devices (Bluetooth ID). Empty if none.

7 Using the platform to manage things and to facilitate the development of context-aware applications

To illustrate the usefulness of the platform and its viability, making the development of context aware applications easier, in this section we describe three sample scenarios where application developers may use the IoT platform to integrate and use information about objects.

7.1 Scenario 1

An application developer wants to monitor the environmental conditions of a greenhouse. He intends to use humidity and temperature sensors and he also intends to visualize the sensors data through a web application.

Without using the IoT platform the developer will have to develop the modules with sensors to collect the data they want to visualize, send the data to the server and then develop modules to store, process and present the data in an appropriate manner. For example involving tables, graphics, view historical data, making customized queries, etc.

Using the platform, the application developer only needs to integrate the Arduino based sensor, e.g. Arduino Yún and DHT22 digital temperature and humidity sensor DHT22, run the standard code for reading DHT22, adjust the message fields to correctly identify the user, the sensor type, location and access level, and send this information to the platform.

Afterward, they are able to use the IoT platform to monitor the data of each sensor using tables or graphs or visualize the data of each sensor between customized periods of dates.

7.2 Scenario 2

An application was developed to be used in a common area of a company that is visited by their workers. The system includes an Arduino based Bluetooth Scanner and a large public display presenting news and advice. The display is connected to a Raspberry pi that receives an URL and displays the correspondent content on the screen. The goal is to present most appropriate content for the visitors at each moment on the display.

The scheduling algorithm is quite simple. Using the user Bluetooth device, to identify and explore the history of information associated to the contents that have been shown for each user in the past, the system should present the content that maximizes the exposure to the current audience, i.e. the system firstly presents the content that a higher number of place visitors have not seen.

Using the platform, the application developer only needs to integrate the Arduino based sensor indicating the platform address and the sensor configuration. Afterward, they get the information about the users that are visiting the place from the platform, using the appropriate API function. Once the algorithm decides which is the next content to be presented, the application sends the platform the information to identify the display (Raspberry pi) and the url that identifies the content. Of course it also need to register the information about users and items presented in the display.

7.3 Scenario 3

In the previous scenario the system adapts its behavior according to the users in the audience. However, unless the system knows the association between the Bluetooth device and its owner, its decisions are based on the fact that the system knows the user's identifier, but it does not know who they really are.

Consider the following scenario:

A company office that has installed two Arduino based sensors. One is an Arduino based digital temperature and humidity sensor, installed within the office. The other is an Arduino based RFID Reader installed near the door.

The application developer integrates the Arduino based sensors in the platform using standard code for these sensors but includes the correspondent configuration for each sensor, i.e. the platform address, the sensor type, the access level and the time period for readings. In this case, there is an identification of the person who carries the identifier. Thus, it is possible to make the association to the worker profile and use this information to select the content (e.g. news) that is according to their interests. Additionally, the developer may ask the platform what the temperature and humidity of the office is and use this information to control the air conditioning.

8 Conclusions and future work

This paper has discussed the development of an open and lightweight Internet of Things platform to facilitate the development of context-aware applications. We have analyzed some previous works with similar goals and we have discussed some issues that need to be addressed when developing an IoT platform. Based on this knowledge, we have proposed and developed an open and lightweight IoT platform that allows application developers and end-users to integrate IoT objects, to create new sensor data models, to monitor and manage objects through the IoT web portal and that further allows developers to access and use, in their applications, high level APIs to handle the information related to the IoT objects. In this approach we place particular focus on two features: simplicity and flexibility. Simplicity, allowing application developers to easily integrate their own objects using their own semantics, and allowing end-users to visualize and manage their objects through the web portal. Flexibility, due its open nature that supports the integration of new types of objects and data formats, it allows developers to customize their own devices and models.

We have performed some initial experiments to evaluate the proposed platform from different perspectives, namely: how easy is it to integrate different objects in the platform; how easy is it to use information from the IoT platform and how it helps in monitoring and configuring Things. The results from these experiments suggest that this is a viable approach to manage Things on the Internet and to facilitate the development of context aware applications, allowing developers to concentrate on the development of their applications and release them from the burden of dealing with the particularities of each sensor and the difficulties of storing and processing context.

As for future work, we plan to evaluate the IoT platform from both perspectives, end-users and context aware application developers in real scenarios. We also intend to extend the platform to support more sources of information (e.g. content from web sources in the Internet), different sensors and actuators and also extend the API (e.g. statistical functions, exploit the historic data in order to derive new information).

References:

1. ITU-T, *Global information infrastructure, internet protocol aspects and next generation networks. Overview of the Internet of things.*, in *Y.2060*. 2012.
2. IERC, *The Internet of Things 2012. New Horizons.* 2012: Halifax.
3. Vermesan, O., et al. *Internet of things strategic research roadmap*. 2009 [cited 2016 April 1]; Available from: http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf.
4. Dempo, H. and M. Yoshida. *CUBIQ: Cross UBIQuitous Platform Architecture*. in *Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on*. 2010.
5. Doukas, C. and F. Antonelli. *COMPOSE: Building smart & context-aware mobile applications utilizing IoT technologies*. in *Global Information Infrastructure Symposium, 2013*. 2013.
6. Mandler, B., et al. *COMPOSE -- A Journey from the Internet of Things to the Internet of Services*. in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. 2013.
7. *Thingspeak*. 2015 [cited 2015 november 3]; Available from: https://thingspeak.com/.
8. *Carriots*. 2015 [cited 2015 november 3]; Available from: https://www.carriots.com/.
9. Perera, C., et al., *CA4IOT: Context Awareness for Internet of Things*, in *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications*. 2012, IEEE Computer Society. p. 775-782.
10. Soldatos, J., et al., *OpenIoT: Open Source Internet-of-Things in the Cloud*, in *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*, I. Podnar Žarko, K. Pripužić, and M. Serrano, Editors. 2015, Springer International Publishing: Cham. p. 13-25.
11. Kim, J. and J.W. Lee. *OpenIoT: An open service framework for the Internet of Things*. in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. 2014.

12. Blackstock, M., et al. *MAGIC Broker 2: An open and extensible platform for the Internet of Things*. in *Internet of Things (IOT), 2010*. 2010.
13. Sicari, S., et al., *Security, privacy and trust in Internet of Things: The road ahead.* Computer Networks, 2015. **76**: p. 146-164.
14. Perera, C., et al., *Context Aware Computing for The Internet of Things: A Survey*. Communications Surveys & Tutorials, IEEE, 2014. **16**(1): p. 414-454.
15. Li, X., et al., *Context Aware Middleware Architectures: Survey and Challenges.* Sensors, 2015. **15**(8): p. 20570.
16. Whitmore, A., A. Agarwal, and L. Da Xu, *The Internet of Things—A survey of topics and trends.* Information Systems Frontiers, 2015. **17**(2): p. 261-274.
17. Bandyopadhyay, D. and J. Sen, *Internet of Things: Applications and Challenges in Technology and Standardization.* Wireless Personal Communications, 2011. **58**(1): p. 49-69.
18. Yen-Kuang, C. *Challenges and opportunities of internet of things*. in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*. 2012.