An Educational IoT-based Indoor Environment Monitoring System

Dario Assante, Claudio Fornaro
Faculty of Engineering
International Telematic University UNINETTUNO
Roma, Italy
d.assante@uninettunouniversity.net, c.fornaro@uninettunouniversity.net

***Abstract* — Internet of Things (IoT) technologies are quickly entering in every aspect of our life, definitely changing the way we interact with smart objects. Such technologies allows to set-up solutions that can be used not just for practical activities, but also for didactic and research. This paper presents the development of an IoT-based environmental monitoring system for indoor workplaces. The remotely accessible monitoring system allows the students to add/remove sensors to the network, configure the nodes and the network, deal with different kind of sensors and controllers, analyze and postprocess the collected data. It also allows to compare the energy consumptions for environmental control with the real environmental wellness. The implemented network and the available didactic scenarios are discussed.**

Keywords — Internet of Things, IoT education, Environmental monitoring

I. INTRODUCTION

Several studies have recently evidenced the importance of indoor environmental monitoring. The World Health Organization as announced potential risks for human health connected to air pollution as worldwide health treat. Such risks are not only in outdoor environments, evidencing even more violation of the air pollution standards also in indoor measurements [1-3]. Additionally, energy consumption for environmental control is still huge: according to the EU Strategy on Heating and Cooling, heating and cooling in buildings and industry accounts for half of the EU’s energy consumption and much of it is wasted. Also, 84% of heating and cooling is still generated from fossil fuels while only 16% is generated from renewable energy. In order to fulfil the EU’s climate and energy goals, the heating and cooling sector must sharply reduce its energy consumption and cut its use of fossil fuels [4]. The EU Strategy on Heating and Cooling considers automation and environmental control on of the key aspects to reduce energy wastes and consumptions. All these reason are increasing the interest towards diffused indoor environmental monitoring systems.

Internet of Things (IoT) technologies are quickly changing the way we interact with object, now smart and able to continuously gather and share huge amounts of data [5-6]. Such smart objects are also able to interact among themselves, share data and autonomously take decisions. Additionally, the IoT-based applications are even smaller, cheaper and with an even lower power consumption, encouraging their wide diffusion. One of the main applications of IoT solutions is for home automation, as natural evolution of domotic technologies. This has been facilitated by the diffusion of smart environmental sensors and simple and efficient control systems. All the gathered data can be collected and post-processed in remotely accessible IoT platforms [7-9].

In this paper, we describe the implementation of an IoT-based internal environment monitoring system, able to control the temperature, pressure, humidity and pollution in different rooms of an apartment. With respect to existing commercial monitoring systems, it has been explicitly oversized in terms of instruments and functionalities, because it is intended to be a system useful both for research and didactic porpoises, more than for just monitoring.

II. THE THINGWORX PLATFORM

The platform used for the implementation of the monitoring system is PTC ThingWorx, being PTC one of the leaders of the market [10]. ThingWorx is a platform that allows the rapid creation of complete solutions for the industrial IoT. It offers the industry's deepest functional capabilities to connect, create and deploy IoT solutions quickly and easily [11].

ThingWorx requires that individual sensors and physical devices be defined and registered on the platform, before data can be transferred or downloaded. Only data from registered devices will be accepted and entered into the platform.

Part of the data management function is to apply the logic and the rules to the incoming data, e.g. checking if there are formulas or algorithms to be applied, verify if calculations are needed or triggers have to be generated according to the condition of the sensor/device. Rules can be different based on the location of the sensor, its type, the type of data produced, etc.

Both physical and virtual data are supported, physical data are just raw data from sensors, while virtual data are created using formulas and transfer functions. By adding formulas to the received data, it is possible to create new data types, values and entities. For example, a device signals temperature and humidity and based on these physical values it is possible to create a virtual value by applying some computation on them. The new virtual value can now be evaluated against other rules and logic and possibly generate another value for further data management.

The *data rules* defined in ThingWorx can trigger an event to be executed. If a value does not fall within a defined range of values, it activates a call to an external system such as a heating device, for example, if the room temperature is low. Using internal features, it can control and manage external systems for cooling, heating, lighting, etc.

Devices and gateways must be configured. For each type of device it is possible to create a corresponding model (template) that defines which sensors must be used to collect data, how many times data must be sent, what are the thresholds for each sensor, etc. Each instance of a device inherits settings from a specific template when it is registered on the platform. It is always possible to override this by setting specific values.

The ThingWorx platform monitors all the connected devices and produces a log to keep their information: an object identifier, the type of the device, its location, the firmware version, etc. Axeda is the built-in component that deals with device management, in particular it allows to remotely monitor and configure the devices in the very moment they are connected to the network, to monitor traffic devices, to track error rates, etc. while automating the delivery of updated firmware and configuration for the devices.

To interact with the platform from a system external to its ecosystem in a safe, simple and effective way, PTC ThingWorx provides a set of REST APIs. *Representational State Transfer* (REST) is an architecture that defines a set of constraints and properties based on HTTP. The Web services that conform to the REST architecture and those between RESTful systems ensure interoperability on the Internet. REST-compatible Web services allow requesting the systems to access and manipulate textual representations of Web resources by means of a uniform and predefined set of operations. It is worth considering that, thanks to a stateless protocol and standard operations, REST systems aim for fast performance, reliability, and growth capacity by reusing components that can be managed and updated without affecting the system as a whole, even while it is running. REST can use messages with format JSON, CSV, HTML, and XML.

III. System Design and Set-up

The solution architecture follows the stack of a classic IoT system, divided into 4 levels:

1. Hardware: level where the devices closest to data collectors are present. In particular, an Arduino with connected sensors and actuators was used.

2. Gateway / Network: Intermediate level between the data collected by the sensors and the platform. A Raspberry Pi connected via WiFi to the platform has been used to manage the exchange of data and inputs to the platform and vice versa, using the AlwaysOn protocol.

3. Platform: Level related to the platform and to all its functionalities. In particular, the ThingWorx PTC platform was used, which offers Device Management services, Data Management and various types of connections both with Edge Devices and with external applications.

4. Application: Application side for the user. Development of a web application to interact with the platform through REST API, allowing monitoring and management of sensors and actuators.

In terms of hardware, the apparently illogical decision to adopt a different solution in each room has been taken. In detail, some of the chosen configurations are

- a breadboard with independent temperature, humidity, pressure, luminosity and pollution sensors, connected to an Arduino board for control;

Fig. 1. System architecture.

- a BME280 sensor, an integrated temperature-humidity-pressure sensor connected to a controller;
- a Climastick board, an on-board environmental sensing already equipped with temperature, humidity, pressure and lux meters and already equipped with WiFi connectivity based on ESP8266;
- a commercial all-in-one environmental meter datalogger (air velocity/temp, light, ambient temperature, humidity).

Also, difference connections are chosen in different rooms, including LAN, Bluetooth and WiFi.

Such architecture is against every rational design criteria: usually, different solutions are compared and then the most suitable one is adopted and replicated in all the spaces. This approach based on standardization ensures an easier management of the systems, reduction of costs and low interoperability problems. However, the proposed network has been implemented not for commercial porpoises and so it doesn't follow the common design criteria. It has been designed as a "technical playground" for students and researchers that want to test different architectures, different kind of meters and different connection methodologies. So, the heterogeneity in the network, despite the interoperability problems, is an added value.

IV. System Implementation

The result of the implementation of the project is a user-friendly App that interfaces to all the devices located inside a museum and performs an analysis on the collected data. The museum manager has been selected as the test user, but the person monitoring the assets security could have been selected as well.

The developed tool is a control panel, which allows the user to perform the following functions:

• Securely authenticating the user

• Monitoring and management the alerts

• Monitoring the microclimate and salt pollution

• Acting the actuators inside the museum

• Analyzing historical and forecast data

• Integrating with a mobile messaging service

Technically this panel is a WebApp created by a REST Web Server, therefore it can be run everywhere on the Internet (if an Internet connection is provided) by using any kind of browser on computers or on smartphones through the REST APIs.

he application allows a secure access through a Login page, the same page provides for user registration. The login and registration window appear as follows:

The login operation is protected by the encryption of the passwords and the ability to save user's credentials. Once logged in, the Application home page is presented, as shown in Fig. 2.

Fig. 2. Platform home page.

At this level, the user is presented a view from above of the entire museum structure with the individual sensors clearly shown. When clicking a specific sensor, a banner is shown with the last data recorded by this sensor.

On the left side of the homepage, and likewise in all the other pages that follow, an *alert column* lists all the "critical" values recorded by the sensors, as shown in Fig 3. In addition to the recorded value, information on the alert such as the type of the event resulting, the location of the sensor, and the date are reported:

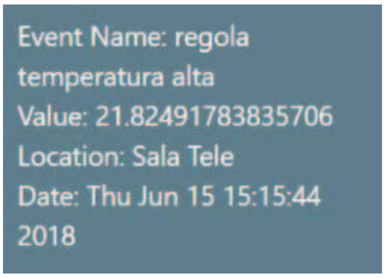

Fig. 3. Alert message.

On the alerts column it is also possible to operate by using the two icons at the top-left: one is to select the room to inspect, the other is to clear the content of the column.

The platform is also able to interact with a "Cisco Spark" real time messaging system through the use of specific APIs. Every time a value outside the predefined range is recorded, an alert is generated and sent to both the web application and a group chat. The idea is to have in this chat all the security and room microclimate monitoring staff in order to quickly react to the problem.

The navigation of the portal is through a series of tabs located above the map of the museum and that allow to enter the details of each room.

Clicking on a tab a new page opens similar to the previous one, in addition this page shows in real-time the main detail of each of the sensors present in the selected room, as shown in Fig 4.

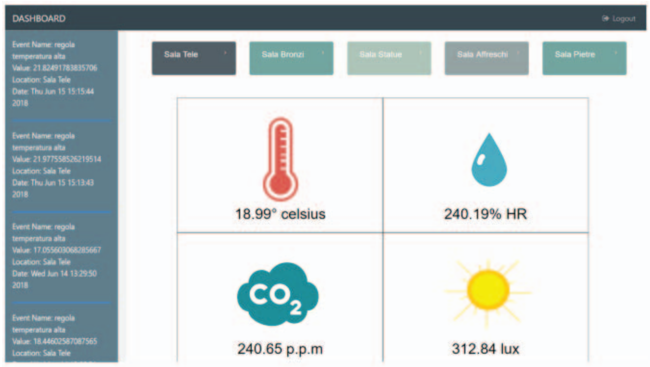

Fig. 4. Environmental data in each room.

Here, by clicking on a specific sensor it is possible to inspect all its details. For example, by selecting the temperature sensor the real-time information shown in Fig. 5 is displayed:

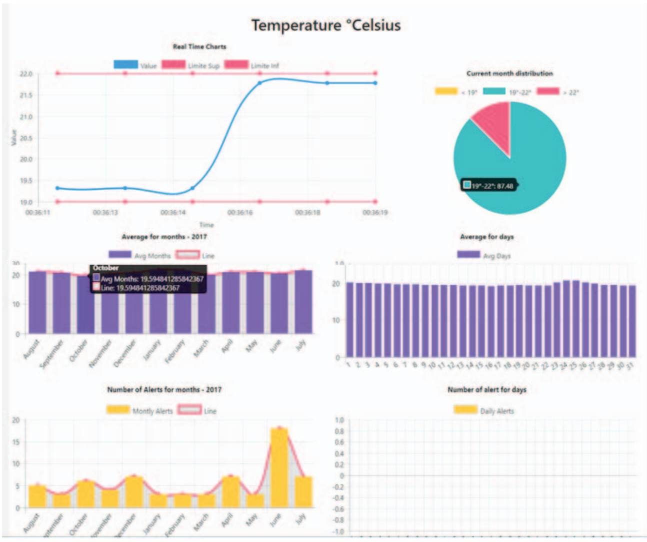

Fig. 5. Real-time statistics.

In this graphic is shown:

1. A linear graphic over the time, with the corresponding upper and lower bounds (when exceeded an alarm is fired), the graphic is interactive to show the value at the position of the mouse pointer (mouseover).

2. A pie chart with the distribution of the values for the current month in three categories: under the lowerbound, over the upperbound, values between the bounds, a mouseover operation allows to inspect every slice value.

3. A histogram with the annual and monthly averages of the measures. A mouseover operation shows the average value for the underlying month. A click operation on the bars allows a “drill-down” operation to show the details of the selected month, day-by-day.

4. The result of the drill-down operation at point 3, a mouseover operation shows the average value for the underlying day.

5. An interpolated histogram shows the annual distribution of the alerts, month-by-month. A mouseover operation shows the average value for the underlying month. A click operation on the bars allows a “drill-down” operation to show the details of the selected month, day-by-day.

6. The result of the drill-down operation at point 5, a mouseover operation shows the average value for the underlying day.

The technologies used to develop the Application are:

Python – interpreted programming language used for the back-end part, its main advantage is the large amount of libraries and frameworks already available.

Flask – micro-framework written in Python used for building the web server as it allows the creation of end-points, it has a small kernel used by third-party libraries to provide database and validation functions. It is based on Jinja 2.

Jinja 2 – template language and engine for Python used to exchange data between back-end and front-end (e.g. the alerts) its markup language.

Sqlite 3 – DBMS used to store sensors data and user information, it is not a client-server application but a collection of functions (libraries) compiled with the application.

SqlAlchemy – toolkit ORM (object-related mapper) for SQL used for mapping database data into programming objects.

HTML5 – the actual version of the HTML standard, it has been used to structure the information to present to the user.

CSS3 – the actual version of the Cascading Style Sheets language, it has been used for the graphic aspect of the HTML pages

JavaScript – scripting language interpreted by the Web browser and used to provide the front-end with dynamic interaction with the user.

Ajax – techniques of Web development client-side to asynchronously interact with the Web sever, it has been used to exchange data with the Pyton back-end.

JQuery –library that simplifies the development of client-side JavaScript scripting, it has been used to ease the management of webpage DOM and Ajax.

Bootstrap – framework used to design web sites and web applications, it has been used for the management of the external graphic of the Application and responsive components.

Chart.js – library that facilitates JavaScript in the visualization of graphics, it has been used to draw the graphics on the dashboard.

V. Educational aspects

The educational purpose can be identified not just in making the hardware design and the software development, but also in performing comparisons and the analysis of the results obtained with different technologies. Moreover the IoT protocols impact on software development and must be taken into account. Security issues are an important topic that must be addressed in every IoT system, however this is a second step and in our demonstrator it has not been taken into account. From the educational point of view, it is important that a real project considers security problems from the first step of the design, as adding security is not always easy and in many cases results in inefficiencies. For this reason, the evolution of our project intend to at least introduce the most important aspects related to protocol security, installation security (and safety also), physical access, denial of services issues. As the project broadens, many different expertise areas must be considered and connected, then teamwork becomes essential. From the educational point of view it is quite formative, as this represents the normal way people works in this field.

VI. Conclusions

An IoT-based environmental monitoring network has been designed and implemented, turning an office in a live laboratory. The network has been designed mainly for educational porpoises, allowing students to access to the network, study the configuration of the different elements of the networks and, in controlled scenarios, to change the network configuration and the connections. It also provides to the students sets of real data, that can be post-processed and analyzed.

The network has been also designed for research porpoises, being the set of real data a basis to monitor the quality of environmental parameters in the offices, but also to estimate the energy consumption and to study strategies for its reduction.

Acknowledgment

This work has been co-funded by the European Commission within the project *IoT4SMEs – Internet of Things for European Small and Medium Enterprises* (2016-1IT01-KA202-005561).

References

[1] S. D. T. Kelly, N. K. Suryadevara, S. C. Mukhopadhyay, “Towards the implementation of IoT for environmental condition monitoring in homes”, IEEE Sensors Journal, vol. 13 (10), pp. 3846-3853, 2013.

[2] C. S. Oh, M. S., Seo, J. H. Lee, S. H. Kim, Y. D. Kim, H. J. Park, „Indoor air quality monitoring systems in the IoT environment”, The Journal of Korean Institute of Communications and Information Sciences, vol. 40(5), pp. 886-891, 2015.

[3] G. Marques, R. Pitarma, “An indoor monitoring system for ambient assisted living based on internet of things architecture”, International journal of environmental research and public health, vol. 13 (11), pp. 1152, 2016.

[4] European Commission, “An EU Strategy on Heating and Cooling”, February 2016.

[5] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, P. Siano, "Iot-based smart cities: a survey", IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), pp. 1-6, 2016.

[6] D. Assante, E. Romano, M. Flamini, M. Castro, S. Martin, S. Lavirotte, G. Rey, M. Leisenberg, M.O. Migliori, I. Bagdoniene, R. Tavio Gallo, A. Pascoal, M. Spatafora, "Internet of Things education: Labor market training needs and national policies", IEEE Global Engineering Education Conference (EDUCON 2018), pp. 1846-1853, 2018.

[7] Bröring, Arne, et al., "Enabling IoT ecosystems through platform interoperability", IEEE software 34.1, pp. 54-61, 2017

[8] J. Guth, U. Breitenbücher, M. Falkenthal, F. Leymann, L. Reinfurt, "Comparison of IoT platform architectures: A field study based on a reference architecture", 2016 Cloudification of the Internet of Things (CIoT), pp. 1-6, 2016.

[9] C. Capasso, O. Veneri, D. Assante, "A novel platform for the experimental training on Internet of Energy", in 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1746-1752, 2018.

[10] The Forrester Wave™: Industrial IoT Software Platforms, Q3 2018.

[11] ThingWorx Industrial IoT.
URL: https://www.ptc.com/en/products/iot/thingworx-platform