An Internet of Things Application with an Accessible Interface for Remote Monitoring Patients

Chrystinne Oliveira Fernandes(✉) and Carlos José Pereira de Lucena

Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil
{cfernandes,lucena}@inf.puc-rio.br

Abstract. E-health area is a research field whose exploration can bring numerous benefits to society. In this paper, we present results from a case study performed in a healthcare environment supported by an Internet of Things (IoT) solution to automate techniques commonly used in patients' treatment and data collection processes. This solution comprises hardware prototypes including sensors, micro-controllers and software agents that work together to make hospital environments more proactive. In addition, the proposed solution provides remote storage of patient data in cloud-based platforms, allowing for any authorized person, including external professionals to work collaboratively with the local team. A web system enables real-time visualization of patient's record as graphical charts through an intuitive interface. Software agents constantly monitor collected data to detect anomalies in patients' health status and send alerts to health professionals when they occur. This work also aims to enable remote patient monitoring to increase proactivity and save resources.

Keywords: Healthcare · Medical systems · Internet of things · Multi-agent systems · E-health · Sensors · Monitoring · Accessibility

1 Introduction

In recent years, the healthcare sector has become one of the main targets for the academic community and financial market. The sector has attracted large investments, mainly because it is a field with great potential for research. The healthcare sector faces many problems such as insufficient resources, high cost of health treatments and the ineffective management of hospital resources. There are also logistical problems such as the amount of patients requiring medical care, a limited number of beds, limited hospital supplies, and few healthcare professionals. Patient monitoring is commonly performed in a reactive way, i.e., medical teams conduct treatment in reaction to the change of the patient's health condition. The reaction time is critical, since the worsening of the patient's condition can result in critical consequences.

Fortunately, these problems are associated with opportunities for improvement, many of them aided by technological resources that we have today. Within this context, our paper presents the results of a case study in a simulated environment aiming to prototype an IoT solution to tackle some of the health problems previously mentioned.

© Springer International Publishing Switzerland 2015

We named this solution Remote Patient Monitoring System (RPMS). The RPMS incorporates the use of smart devices and different software techniques in order to save resources. The general goal of this work is to contribute to a more effective management of hospital resources. The developed prototype provides an infrastructure with an accessible interface aiming to make the hospital environment more proactive, responsive and interactive. We present a technology that acts on the environment and equips it with intelligent devices.

The prototype is based on the Internet of Things (IoT). IoT is field within Computer Science that has grown substantially over recent years. It broadens the established concept of the internet, from a network that connects people to a wider network that allows for the connection between people and things, and between things, with little or no human intervention. Within the IoT context, "things" are real-world entities that are part of the IoT network. These things can vary from simple objects, such as a chair, appliances or cars, including living entities, for instance, plants, animals and persons. Our solution adopts a modeling strategy based on reactive and cognitive software agents to build the IoT application.

The remainder of this paper is organized as follows: Sect. 2 describes the theoretical groundings, discussing aspects of Ubiquitous Computing (UbiComp), IoT, Human-Computer Interaction (HCI) and Multi-Agent Systems (MAS). Section 3 describes the research method used in this study. Section 4 presents the details of the conducted case study. Section 5 concludes this paper with final remarks and future work.

2 Theoretical Groundings

This project has a multidisciplinary approach, involving a variety of research areas, such as the IoT, UbiComp and HCI. Building on these areas, we designed and constructed an e-health system to assist remote patient monitoring activity.

Before discussing the solution in detail, we present a brief overview of fundamental concepts that support the features in this work.

2.1 Ubiquitous Computing

Ubiquitous Computing (UbiComp), as defined by Mark Weiser in the early nineties [REF], relates to the concept where computing power is distributed across devices. In fact, back when Weiser coined the term Ubiquitous Computing, it represented a change of view in Computer Science. The focus changed from the machines to the human. He envisioned that computing power rather than being concentrated in supercomputers with high cost, size and power would be distributed into smaller and cheaper computing devices spread around the environment

According to Weiser, the notion of ubiquity arose from the observation of things in the real world, specifically, the office equipment used in daily activities. Weiser conceived a different approach for these computing devices based on this observation. Instead of modelling the diverse devices' functionalities within a single computer, he proposed transferring the computational power to these devices. Therefore, they would be able to process information provided to them.

According to Weiser, "ubiquitous computing it the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user" [2].

2.2 Internet of Things

Although the early works in Ubiquitous Computing promoted the distribution of computational power, devices were still limited because they were dependent on manual manipulation by people for making data available. These devices produced a large amount of information, but humans generally have limited time resources. This has led to a bottleneck in the process of sharing this information. This way the concept of IoT has emerged, in which things would be able to connect to the Internet and publish its information with little or no human intervention.

Considering things as a new autonomous element of network models represented a paradigm shift. It was possible to automate costly human activities, such as collecting and publishing data over the network. The IoT approach delegates these activities to things, which can interact with each other, with people and systems. Thus, interactions in the network, which before were restricted to human-to-human or human-to-computer, were extended to support other interactions like human-to-thing and thing-to-thing.

One can define IoT as a global network of smart devices that can sense and interact with their environment using the Internet for their communication and interaction with users and other systems [5]. IoT application developers may rely on a variety of communication technologies. In this work, we use RFID (Radio-Frequency IDentification), Arduino micro-controllers and sensors.

RFID is an automatic identification technology based on radio signals, used for sensing and communication [1]. A RFID system consists of two components: the transponder, which is located on the object to be identified and the interrogator or reader, which, depending on the technology used, may be a read-only or read-write device. The reader and transponder are the main components of every RFID system [8].

Arduino, is a modular open-source prototyping platform [9]. It consists of a micro-controller that can be programmed to process input and output of external electronic components connected to it. It is an embedded computing device that allows to build systems to interact with the environment through hardware and software [4].

Developers can use different types of sensors to collect useful data in IoT applications in the e-health context. These sensors can provide a variety of vital data, including heartbeat rate, blood glucose (to monitor diabetic patients), temperature, humidity, liquid level, oxygen level, carbon dioxide level, patient activities (with an accelerometer) and many others.

2.3 Multi-Agent Systems

According to Wooldridge, an agent can be defined as "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives" [6]. Although there is no universal definition for the term agent, there is a consensus in the literature that autonomy is a key feature to the notion of agency. In this context, the possibility of autonomous acting means that no intervention by humans or others systems is necessary for agents to decide how to perform their activities, although the set of possible actions must be set beforehand. Another relevant property of agents is learning, that is, the ability to learn from their experience.

Agents control their internal state and behavior, but they do not have full control over their environment. They have a set of actions to perform that may result in environment changes, so they may influence their environment according to the action they decide to perform. Agents autonomously decide what to do to meet their goals.

2.4 Human-Computer Interface

As defined by Preece, Rogers and Sharp, "HCI is a multidisciplinary field that is concerned with the design, evaluation, and implementation of interactive computing systems. It encompasses all aspects related to the interaction between people and computers" [7].

The quality of interactive systems involves properties such as usability, accessibility, communicability and user-experience. In this paper, we explicitly addressed the accessibility property. According to Tim Berners-Lee, the power of the Web is in its universality and the access by everyone regardless of disability is an essential aspect [10].

According to the W3C's Web Accessibility Initiative, web accessibility means that people with disabilities can perceive, understand, navigate, interact and contribute with the Web [10]. The design of the web system that is part of the RPMS solution considers some accessibility requirements. Using tools developed for visually impaired users, namely screen readers, these users are able to interact with our solution. Screen readers interpret textual content in web pages and convert them to audio format.

In terms of usability, the system supports the use of wearables, bracelets with RFID tags, as a way to interact with the system. For instance, health professionals can identify patients and their records simply by approaching a mobile device, such as tablet, smartphone, or other device with RFID reader to the patients' bracelet. The reading event triggers the display of their data through a web page, containing the medical records history and real-time status.

3 Methodology

In order to perform an in-depth analysis and clarify research challenges regarding patient monitoring through an IoT view, we have defined research questions followed by a case study to explore design and implementation issues.

3.1 Research Questions (RQ)

The following research questions were defined to tackle aspects from monitoring patients' domain that, in our point of view, are those that can bring improvements to this application domain.

RQ1. How can the use of micro-controllers, sensors and wearable devices make patient monitoring processes more proactive?

RQ2. How can the collection and distributing of real-time patient data support patient monitoring?

3.2 Hypotheses (H)

Before the empirical study, we formulated five hypothesis. At the end of this study, these hypotheses were contrasted with data obtained to be confirmed or refuted.

H1. It is feasible to use wearables devices in order to identify patients through an IoT solution

H2. One can collect patients' vital data automatically using an IoT application

H3. It is possible to detect anomalies in patients' health status in real time

H4. The patient monitoring can be done in remote mode

H5. The use of micro-controllers, sensors and wearable devices can make patient monitoring processes more proactive

4 Case Study: Remote Patient Monitoring

4.1 Problem Definition

Scientific research within the context of healthcare is a complex task. Dealing with such sensitive issues as patients' health and the risks involved in their medical treatment encompass challenges of diverse natures. Consequently, technological advances emerge more slowly than in other areas. Barriers to conducting research in this area are not limited to strictly technical matters. The direct involvement of patients in any scientific research demands the definition of an ethical protocol, which a committee may decide to approve or not. Moreover, healthcare staff and patients may be involved in design stages of a software development process, since their perspectives might contribute with practical design issues. However, these parties may be unavailable or may have little spare time for collaborating with the research.

We observe that in many cases the patient monitoring is done reactively. In this case, the medical team only takes action after detecting a decline in the patient's health. We define the term Anomaly Detection Interval (ADI) as the time interval between the moment that the patient had complications in their health status and the time of its detection by health professionals. We define as anomalies noticeable deviations in patients' health status, which requires professional intervention at the instant they occur.

The local monitoring is one of the factors that contribute to raising the ADI, since the healthcare provider needs to displace himself to the location of patients and then collect vital data for the assessment of their condition.

4.2 Proposed Solution

The RPMS is an IoT solution for patient monitoring that supports automatic collection and transferring of patient data to remote repositories. A web system provides access to this data through a user-friendly interface. This system makes use of software agents for anomaly detection, resource negotiation and dynamic reconfiguration in response to changes in the environment. Additionally, the solution helps to visualize patient data remotely stored.

4.3 Goals

We defined the following goals for guiding the development of the prototype solution, aiming to minimize the ADI:

- To reduce the displacement of health professionals in reactive monitoring;
- Automate the process of collecting vital patient data;
- To allow remote data visualization, promoting the dissemination of patient data for medical staff and external collaborators;

4.4 Results

RPMS Architecture. The RPMS is structured in three layers:

L1. Data Distribution Layer: Our solution uses a cloud-based platform to distribute data. This platform is a remote repository of patient data

L2. Data Communication Layer: Performs the communication of the IoT application (L3) with the cloud-based platform (L1) through a REST API. The application sends and retrieves data from the cloud-based platform through HTTP requests

L3. Data Management Layer: Comprises the IoT application with its six modules. It is responsible for the entire information processing in RPMS, ranging from identification, collection, storage, visualization and data monitoring to the dynamic reconfiguration of the system

Figure 1 shows an overview of the architecture of the proposed solution.

RPMS Infrastructure Elements and Technologies Used. To provide the data distribution in our solution, we use a free remote data storage service called Parse. Developers can easily configure applications' backend with this service. This cloud-based platform offers functionalities for saving data objects, file storage, push

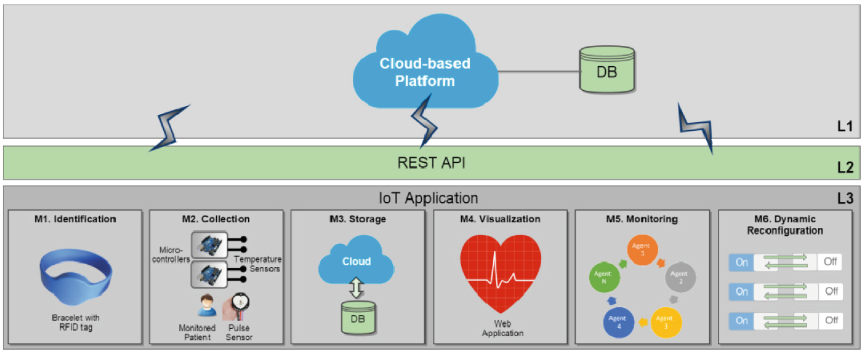

Fig. 1. The RPMS architecture with its three layers (L1-L3). The IoT Application (L3) interacts with a cloud-based platform (L1) through a REST API (L2).

notifications and user management [12]. In addition, it allows data portability, offering a way to export an application's entire data set, in the same format as the REST API. The Parse also has a feature to import data in the REST API format. The platform also has an importer, which supports JSON and CSV files. However, the platform recommends importing data in the REST API format due to its flexibility.

We use a technology called Temboo in order to enable our IoT application to interact with the cloud-based platform. The Temboo is a code virtualization platform. It is used to facilitate building applications and interacting with various APIs [13]. The Temboo has a specific library for Parse, called Parse library, with seven packages available to support code creation for consume the services offered by this cloud platform. In our solution, we are using Queries package to insert and retrieve Parse data. Thus, our IoT application uploads the patient data to the remote data repository using Parse library from the Temboo API. In addition, the application consumes this data in real time and displays them graphically on a web page.

We also use another Temboo library called Twilio to send alerts to health professionals via SMS. Twilio is a communication platform that provides web-services APIs, allowing users to build their own scalable, reliable applications for voice communication and SMS. Different solutions use Twilio, including in the healthcare context [14]. Software developers can use their web services API to make and receive calls and send and receive text messages. Twilio services are accessed via the HTTP protocol. In our application, we use Twilio service, in a simplified manner by the Twilio library from Temboo. Our application uses the SMSmessages package that allows sending and receiving SMS messages [13].

Finally, to provide the functionality corresponding to the RPMS's L3 layer, we are using some hardware and software elements. For the identification of the patient (M1 module in the IoT application), we use an RFID system. This system includes a wearable device corresponding to a bracelet with an RFID tag (Fig. 2) and an RFID reader (Fig. 4-b). The data collection module (M2), in turn, performs the patient data sensing through a hardware prototype containing an Arduino microcontroller with wireless and Ethernet network interface, a temperature sensor and a heartbeat sensor in addition to a protoboard and some male-male type jumpers (Fig. 4-a).

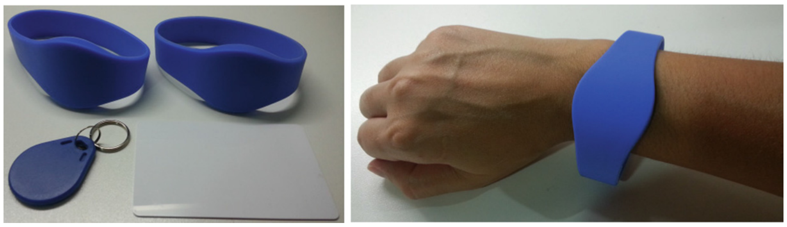

Fig. 2. Accessories that can be used for patient's identification, containing RFID tags: bracelets, card and keychain.

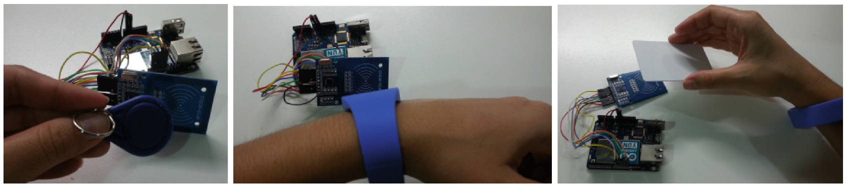

Fig. 3. Readings of accessories with RFID tags through our hardware prototype

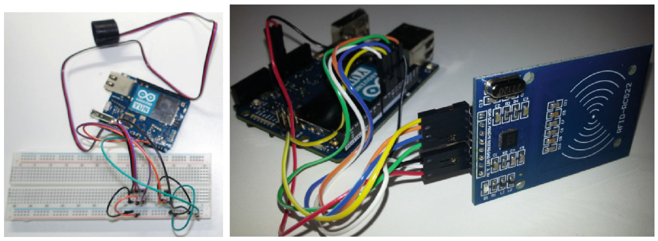

Fig. 4. **(a).** Prototype for monitoring patients containing temperature and pulse sensors. **(b).** Prototype for identification of patients through RFID automatic identification technology.

We used the Arduino Integrated Development Environment (IDE) to implement the M1 and M2 modules in the C ++ programming language. It comes with a set of libraries to build programs easily. Professionals such as designers and artists that are not professional programmers also can build systems easily using Arduino. Some sensors require the installation of external libraries, provided by their own manufacturers as the pulse sensor library used in our solution. This library can be downloaded from the manufacturer's website, along with a getting started guide [15]. We also installed the Temboo library available for download on this API's website, along with its installation manual [13].

We developed the other application modules, M3, M4, M5 and M6 in Java language. To implement software agents in the Monitoring module (M5), we used the JADE (Java Agent Development Framework) framework [11].

RPMS Model. The following table lists the elements that represent the concept of things involved in the environment in which the patient is being treated:

Thing1.	Patients;
Thing2.	Bracelets with RFID tags;
Thing3.	Health Professionals;
Thing4.	Patient's environments (Apartment, intensive care unit, Surgery room)The follow agents were modeled in the system:
Agent1.	DataVerifierAgent: Checks for anomalies in the sensored data;
Agent2.	SMSAgent: Notifies anomalies, by sending messages;
Agent3.	ResourceNegotiatorAgent: Performs negotiations with other agents to get resources;

RPMS Features. The tool is composite for six modules as follows:

M1. Patient identification: Patient identification: Performed by an RFID system, where the patient uses a bracelet with an RFID tag in order to be identified it in the system by a health professional. When health professionals read a patient's bracelet using a device such as a smartphone or a tablet, the data of this patent can be quickly visualized in their device. The RFID readings triggers the initialization of a web page with patient data (Fig 3).

M2. Collection: our application collects automatically patient data such as pulse and body temperature. This collection process is made using a hardware prototype comprising sensors and micro-controllers. The application also can collects information about patient's environment. This information can be light, humidity and noise level of environment.

M3. Storage: Once collected, the system transmits the patients' data to a cloud database service if connectivity is available. Otherwise, the application persists data locally and transmits them later on.

M4. Visualization: A web application visualizes the patient records. Any authorized person can access it from multiple devices (computer, smartphone or tablet). This application has a user-friendly interface with accessibility features for people with visual impairment. The application requests data from the cloud platform and exhibits at the web page in line charts. The plotted data is updated in real-time, in interval time settings previously. The Fig. 5 shows an example of a chart corresponding to patient temperature data.

M5. Monitoring: This module constantly monitors the collected patient data through agents that should react in case of anomalies detection. Reactive and cognitive software agents support this detection mechanism. The system has a mandatory configuration step for each patient as follows: an administrator user registers and defines the Desired Value Range (DVR) for each sensor (e.g., expected temperature and pulse values).

We call the values that are outside this range Anomalous Values (AV). The values that are close to the minimum and maximum values are associated with abnormalities. Each of these anomalies receives a representative label within the healthcare context, so

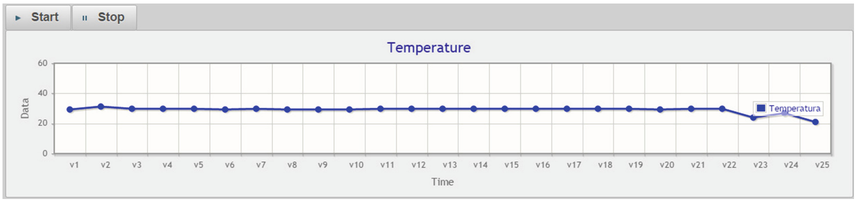

Fig. 5. Charts corresponding to the sensor readings from patients and its environment

that it makes sense to a domain expert. The goal is to enable a healthcare professional to quickly identify any problem occurring when the anomaly is detected by the system. In addition, the administrator must assign a professional to handle each anomaly, who will be alerted of its occurrence. This notification can be through e-mail or SMS. Software agents are in charge of this continuous detection of minimum and maximum anomalies, as well as sending alerts to professionals.

Cognitive agents use argumentation techniques to work collaboratively on shared resources, promoting a more effective resource management. The system outputs recommendations, such as which patient should be next in line for medical care. The agents also suggest which patient needs to receive, more urgently, medical assistance like a nursing professional to administer their medication.

M6. Dynamic reconfiguration: The collection module is context sensitive, i.e., the system can react dynamically in response to changes in the environment. The displacement from a patient to a new environment can invalidate the defined data range for the anomalies. In this case, the application will need to be reconfigured. The user can reconfigure them through the application interface or the system can reconfigure itself autonomously by intelligent agents. In the latter case, the agents will exchange information, so that new agents can learn the environment configuration, by querying other agents that already know the environment conditions.

5 Conclusions and Future Works

We can observe that the proposed solution makes the environment more proactive, since the system detects anomalies in real time and sends alerts autonomously through software agents. Responsible health professionals for handle those anomalies can take action immediately in response to these events. A possible indication of proactivity can be verified by measuring the DAI.

One can note a large human-to-thing interaction in the system, since agents (considered things in this application) are able to alert professionals about anomalies in the patients' health status. Additionally, there is a large thing-to-thing interaction, since VerifySensorDataAgent and SendSMSAgent agents, can communicate with each other effectively for achieving their goals.

One can see that this solution make the patient's identification process more efficient, since it replaces the traditional identification techniques with RFID identification technologies.

Our solution also improves the process of collecting vital patient data, since it replaces the manual data collection processes, which is more prone to errors.

The patient data distribution over the Internet allows any authorized person to consult them from anywhere at any time. This functionality also brings benefits, such as the possibility of different experts to collaborate and discuss about the patient's health status and possible treatments.

As future work, we consider using cognitive agents for performing activities such as recommendation of possible treatments and medications and predicting potential patient's health conditions that can lead to anomalies. In addition, we plan to provide remote monitoring already in ambulances during the patient's way to the hospital.

Acknowledgements. This work was supported by grants from CAPES.

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. J. Comput. Netw. Int. J. Comput. Telecommun. Network. **54**, 2787–2805 (2010)
2. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**(7), 75–84 (1993)
3. Kuniavsky, M.: Smart Things Ubiquitous Computing User Experience Design. Elsevier, Amsterdam (2010)
4. Norvig, P., Russell, S.: Artificial Intelligence (in Portuguese). Elsevier, Amsterdam (2013)
5. Doukas, C.: Building Internet of Things with the Arduino. CreateSpace Publisher, North Charleston (2012)
6. Wooldridge, M.: An Introdcution to MultiAgent Systems. Wiley, New York (2009)
7. Preece, J., Rogers, Y., Sharp, H.: Interaction Design, 1st edn. Wiley, New York (2002)
8. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 2nd edn. Wiley, New York (2003)
9. Arduino. http://arduino.cc/
10. W3C Web Accessibility Initiative. http://www.w3.org/WAI/intro/accessibility.php
11. JADE. http://jade.tilab.com/
12. Parse. https://parse.com/
13. Temboo. https://www.temboo.com/
14. Twilio. https://www.twilio.com/
15. PulseSensor. http://pulsesensor.com/pages/code-and-guide