UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

Intelligent Personal Assistants Solutions in Ubiquitous Environments in the Context of Internet of Things

João Filipe Lopes dos Santos

Submitted to the University of Beira Interior in candidature for the Degree of Master of Science in Informatics Engineering

Supervised by Prof. Dr. Joel José Puga Coelho Rodrigues

Department of Informatics
University of Beira Interior
Covilhã, Portugal
http://www.di.ubi.pt

This work has been partially supported by TIMWE *Investigação e Desenvolvimento*, through the AMBRO project, co-funded by COMPETE/QREN/EU, in the context of the *Portuguese Sistema de Incentivos à I&DT* (project code: 33779), by Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project.

COMPETE
PROGRAMA OPERACIONAL FACTORES DE COMPETITIVIDADE

QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
PORTUGAL 2007.2013

UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

Acknowledgements

This dissertation marks the end of another stage of my academic career, concluding this way my Masters Degree. The support of every person that contributed, directly or indirectly, to my work throughout the last year was very important to achieve this goal. I am most grateful to every one of them.

I would like to start to thank to my family, especially to my parents, José and Maria, and my sister Catarina. Only their devotion and hard work made my success in academic life possible, and it is something that I will never forget in my entire life. After all, I am what I am today thanks to them.

I express my gratitude to my supervisor, Prof. Dr. Joel José Puga Coelho Rodrigues, for supervising me throughout my dissertation, and suggesting me to work on such a challenging and interesting project. I also have to thank him for giving me the chance to join the Next Generation Networks and Applications Research Group (NetGNA). Inside the NetGNA, I have to give a special thanks to Bruno Silva, who helped me along my work on this dissertation.

I am also very grateful to TIMWE I&D for giving me the opportunity to work with them on their project. It was a very enriching experience for me, giving me more expertise and knowledge to my future work. It is with pleasure that I would like to thank all the people that I worked with on the TIMWE Lab, especially to João Casal, since they helped me a lot to achieve my goals on this project.

Finally, I would like to express my gratitude to all my great friends who supported me in good times and bad times over this past year.

Abstract

Internet of Things (IoT) will create the opportunity to develop new types of businesses. Every tangible object, biologic or not, will be identified by a unique address, creating a common network composed by billions of devices. Those devices will have different requirements, creating the necessity of finding new mechanisms to satisfy the needs of all the entities within the network. This is one of the main problems that all the scientific community should address in order to make Internet of Things the Future Internet.

Currently, IoT is used in a lot of projects involving Wireless Sensor Networks (WSNs). Sensors are generally cheap and small devices able to generate useful information from physical indicators. They can be used on smart home scenarios, or even on healthcare environments, turning sensors into useful devices to accomplish the goals of many use case scenarios.

Sensors and other devices with some reasoning capabilities, like smart objects, can be used to create smart environments. The interaction between the objects in those scenarios and humans can be eased by the inclusion of Intelligent Personal Assistants (IPAs). Currently, IPAs have good reasoning capabilities, improving the assistance they give to their owners. Artificial intelligence (AI), new learning mechanisms, and the evolution assisted in speech technology also contributed to this improvement. The integration of IPAs in IoT scenarios can become a case of great success. IPAs will comprehend the behavior of their owners not only through direct interactions, but also by the interactions they have with other objects in the environment. This may create ubiquitous communication scenarios

where humans act as passive elements, being adequately informed of all the aspects of interest that surrounds them.

The communication between IPAs and other objects in their surrounding environment may use gateways for traffic forwarding. On ubiquitous environments devices can be mobile or static. For example, in smart home scenarios, objects are generally static, being always on the same position. In mobile health scenarios, objects can move from one place to another. To turn IPAs useful on all types of environments, static and mobile gateways should be developed. On this dissertation, a novel mobile gateway solution for an IPA platform inserted on an IoT context is proposed. A mobile health scenario was chosen. Then, a Body Sensor Network (BSN) is always monitoring a person, giving the real time feedback of his/her health status to another person responsible by him (designated caretaker). On this scenario, a mobile gateway is needed to forward the traffic between the BSN and the IPA of the caretaker. Therefore, the IPA is able to give warnings about the health status of the person under monitoring, in real time. The proposed system is evaluated, demonstrated, and validated through a prototype, where the more important aspects for IPAs and IoT networks are considered.

Keywords

Internet of Things, Intelligent Personal Assistants, ubiquitous computing, mobile gateway, Body Sensor Network, mobile health

Contents

List of Figures

List of Tables

Acronyms

6LoWPAN: IPv6 over Low Power Wireless Personal Area Network
AI: Artificial Intelligence
API: Application Programming Interface
ART: Android Runtime
AVT: Angular Velocity Threshold
BLE: Bluetooth Low Energy
BSN: Body Sensor Network
CBR: Case-Based Reasoning
CoAP: Constrained Application Protocol
DAG: Directed Acyclic Graph
DODAG: Destination Oriented Directed Acyclic Graph
DTLS: Datagram Transport Layer Security
DVM: Dalvik Virtual Machine
ECG: Electrocardiogram
EMG: Electromyography
GCM: Google Cloud Messaging
GPRS: General Packet Radio Service
GPS: Global Positioning System
GSR: Galvanic Skin Response
H2M: Human-to-Machine
HMM: Hidden Markov Model
HTTP: Hypertext Transfer Protocol
IBM: International Business Machines
IEEE: Institute of Electrical and Electronics Engineering
IETF: Internet Engineering Task Force

iOS: iPhone Operating System

IoT: Internet of Things

IP: Internet Protocol

IPA: Intelligent Personal Assistant

IPDA: Intelligent Personal Digital Assistant

KM: Knowledge Management

LAT: Lower Acceleration Threshold

LLN: Low Power and Lossy Network

LoWPAN: Low Power Wireless Personal Area Network

LR-WPAN: Low Rate Power Wireless Personal Area Network

LTE: Long Term Evolution

MAC: Media Access Control

M2M: Machine-to-Machine

MAS: Multi-Agent System

MQTT: Message Queue Telemetry Protocol

MQTT-S: Message Queue Telemetry Protocol for Sensor Networks

MTU: Maximum Transmission Unit

NLUI: Natural Language User Interfaces

OMT: Object Modelling Technique

OS: Operating System

PDA: Personal Digital Assistant

QoS: Quality of Service

REST: Representational State Transfer

RFID: Radiofrequency Identification

ROLL: Routing over Low-Power and Lossy Networks

RPL: Ripple Routing Protocol

SAS: Single-Agent System

SIP: Session Initiation Protocol

SLAAC: Stateless Address Auto Configuration

TCP: Transmission Control Protocol

TTS: Text-to-Speech

UAT: Upper Acceleration Threshold

UDP: User Datagram Protocol

UML: Unified Modelling Language

URI: Universal Resource Identifier

USB: Universal Serial Bus

VoIP: Voice over Internet Protocol

WBSN: Wireless Body Sensor Network

WG: Working Group

WSN: Wireless Sensor Network

WWW: World Wide Web

XML: eXtensible Markup Language

XMPP: Extensible Messaging and Presence Protocol

XEP: XMPP extension protocol

1. Introduction

The first chapter presents an overview of the most important concepts for this dissertation, including its main focus, objectives, contributions for the state of the art, and organization.

1.1. Focus

The importance of technology in human life increased along the last decades. Many processes in different business areas were automated through the use of technological solutions. The world has been transformed in a symbiotic environment between humans and machinery.

One of the most important achievements in the technological world was the creation of the Internet. It all began in the 1960s, when Leonard Kleinrock published the first paper on packet switching theory [1]. His queuing theory demonstrated the theoretical efficiency of the packet switching communication, serving as a basis for the creation of the ARPANET – Advanced Research Projects Agency Network [2]. The ARPANET, first tested in 1969 when two computer nodes established a communication, was initially used for military and research purposes only, and can be seen as the precursor of the actual Internet.

The evolution of the Internet to become the global communications network that is known today has been occurring over the years with the development of new paradigms and technologies, such as the Transmission Control Protocol/Internet Protocol Suite (TCP/IP). TCP/IP was officially standardized in 1983, and is the computer networking model used nowadays in the Internet.

The importance of the Internet to the human society, in general, started to be fully noticed in the 1990s with the creation and commercialization of the World Wide Web (WWW). It has revolutionized communications and social networking, allowing new forms of business, like the e-commerce, new forms of communication, like the e-mail or Voice over Internet Protocol (VoIP) calls, and new forms of accessing information, like blogs, newspaper sites, and Internet search engines like Google. The creation of new services was enabled, being accomplished through the use of Internet enabled devices. Those devices treat big data sets of information at a much faster pace when compared with the speed at the humans can do it. That is one of the reasons that make the creation of ubiquitous communication scenarios an advantageous topic for the human society [3].

The creation of ubiquitous communication scenarios can be enabled by the Internet of Things (IoT) technology. The IoT defines that almost every object in the surrounding environment can have an IP address, being in this way identifiable, addressable, and consequently connected to the Internet [4][5].

With the creation of IoT networks, the amount of data generated increases in an extraordinary way, with the number of devices generating traffic being bigger than before. This data can enable the creation of new types of services and applications that can be used to fulfill many of human life requirements. IoT can create a big impact in the business by automating some processes and improving the control of many environment variables easing the decision making process on businesses [6].

There already exist some solutions that use IoT scenarios. One example is the creation of smart home environments, like the one shown in [7]. It demonstrates that is possible to integrate Web Services, cloud computing and the IoT to create scenarios on which smart objects can autonomously make decisions inside our houses. Another great opportunity that IoT technology enables is the smart city vision. A smart city is an urban infrastructure able to interact, in real time, with its surrounding environment in an autonomous and insightful manner. Using low-power sensors, wireless networks, Web and mobile-based applications, smart cities aim to use better the public resources, offering greater services to citizens at lower operational costs. One example is the “Padova Smart City” project presented on [8] ,where some indicators about the environment are measured and controlled.

IoT can have a great impact on healthcare. As presented in [9] and [10], the use of biomedical sensors in IP networks can help doctors monitoring their patients. Making use of some specific technologies designed for constrained devices, ubiquitous healthcare systems can be deployed. On those systems, vital signals from patients are sensed by biomedical sensors inside a wireless sensor network (WSN). These types of deployments can improve the doctors’ work, by enabling the remote access to the medical records of the patients through mobile devices (like PDAs).

The Manufacturing business can also benefit from the use of IoT technology. For example, in [11] and [12], remote manufacturing monitoring systems are described. These systems can gather information about industrial machines, helping to manage and control the business. The collection of data from industrial equipments can be done by combining the use of sensors (temperature, humidity, vibration, pressure, among others) with radio frequency identification (RFID) technology, ZigBee, WiFi and Ethernet networks.

The technologic evolution not only allowed the evolution of networks. New types of devices also appeared. One example is the creation of Intelligent Personal Assistants (IPAs). IPAs are mobile, autonomous, software agents capable of performing tasks or services on behalf of humans [13].

An IPA is a multitasking machine where a simple user command may trigger multiple processes at the same time. Using its learning mechanism, and knowing its context, an IPA can give an autonomous response to a user request, enhancing its capability to learn from repetitive actions.

A way to improve the learning mechanism of IPAs would be the integration of information generated from the surrounding environment, like temperature, humidity, luminosity, among others. These external indicators could be measured with sensors that, in some way, would transmit the gathered values to an IPA. Those values can be treated later by IPAs to trigger services able to assist humans in their daily lives. All this process can be seen as the natural interaction between people, environment and machines, creating a scenario of ubiquitous computing.

IPAs can be used for different purposes. Smart home, healthcare, and other business areas can benefit from the use of IPA devices. In the healthcare field, intelligent health monitoring systems can be deployed by enabling the connection between Body Sensor Networks (BSNs) and IPAs. A caretaker (being a physician, a familiar, or a friend), can be informed about the condition of a person under monitoring in real time by receiving notifications from his IPA. This is due to the ability that IPAs have to understand if the information coming from the sensors inside the BSN is normal or not (usual heart rate, blood pressure, among others). The integration of IPAs in IoT scenarios can accelerate the creation of intelligent and autonomous systems.

1.2. Problem Definition

Billions of devices can exist on ubiquitous communication scenarios. Those devices generate a great amount of traffic that can have any destination. The traffic can be directly forwarded if the recipient device is on the same network of the emitter. If the recipient is on another network, a gateway device is needed to forward the traffic.

The uncertainty about the type of environments where the devices are inserted raises some issues. If a device is on a static environment it means the device is on a confined position, being always reachable by the gateway. On such scenarios static gateways can be used. On scenarios where devices are constantly moving, like mobile health environments, static gateways cannot be used. Therefore, the gateway should be performed by a mobile device.

The utilization of IPAs on ubiquitous communication scenarios inherits the traffic forwarding problem already explained. On mobile environments, smartphones can be used as mobile gateways. But it is necessary to verify whether devices like smartphones are completely able to play the mobile gateway task, since new smart power consumption mechanisms and other relevant performance evaluation strategies may need to be deployed. With all these addressed questions, IPAs can be used to remotely monitoring health status of persons on a society which is getting increasingly older [14]. This is just one of the possible solutions where the integration of IPAs in IoT scenarios can improve a person's life.

1.3. Objectives

The main objective of this dissertation is the presentation of a novel mobile gateway solution for an IPA platform inserted on ubiquitous environments on an IoT context.

The IPA platform created, called AMBRO, is an infrastructure able to insert an IPA on ubiquitous communication scenarios. The AMBRO personal assistant is supposed to be a powerful tool for its users, giving advices and assisting them in many of their daily life tasks.

The AMBRO mobile gateway application was developed with the objective of allowing an AMBRO owner to use it as a health status-monitoring tool. To do that, the AMBRO mobile gateway application includes three monitoring services, each one controlling one sensor on a BSN. This BSN is based on a global positioning system (GPS) sensor, a heart rate sensor, and a sensor module composed by an accelerometer and a gyroscope to detect a possible fall of its user. The AMBRO mobile gateway application, installed on an Android smartphone, is able to forward the traffic generated on the BSN to the AMBRO cloud, and vice-versa. This application allows a person to be monitored anytime and anywhere, allowing another person (usually designated by caretaker) on another place to receive the information relative to each monitoring service on its IPA.

To accomplish the main objective, the following intermediate objectives were identified:

- Study of the state-of-the-art about the IoT, IPAs, and the solutions combining IPAs and the IoT;
- Identification of the technologies that can enable the insertion of IPAs on ubiquitous communication environments;
- Definition of the requirements necessary to deploy a IPA;
- Analysis of the system requirements necessary to create a mobile gateway solution for an IPA platform;
- Proposal and deployment of a novel mobile gateway solution for an IPA platform on an IoT context, including a mobile application for the Android operating system (OS);
- Insertion of the mobile gateway on a mobile health scenario comprised by heterogeneous devices;
- Generate useful information for the IPA through the data gathered from the BSN;
- System demonstration, performance evaluation, and validation by experimenting the whole system, comprising the AMBRO platform, the AMBRO mobile gateway and the BSN, using multiple devices.

This research expects to produce the desired application, providing a novel engineering solution capable of disseminating significant knowledge for the worldwide scientific entities.

1.4. Main Contributions

This section presents the scientific contributions of this dissertation to the state-of-the-art involving IPAs inserted on IoT contexts.

The first contribution involves a novel mobile gateway solution for ubiquitous mobile health scenarios, in which information gathered from a BSN is used by an IPA belonging to a caretaker. This solution is presented in a paper entitled "Internet of Things Mobile Gateway Services for Intelligent Personal Assistants" submitted to the IEEE HEALTHCOM 2015, Boston, MA, USA, October 2015.

The second contribution of this dissertation is a survey about the state of the art of IPAs solutions based on IoT networks. A paper with this contribution entitled "Intelligent Personal Assistants based on Internet of Things Approaches" was submitted to the IEEE Systems Journal by August 28th, 2015, and it is under review.

The third contribution is a paper entitled "A IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments", which presents a detailed explanation of the mobile gateway system for ubiquitous mobile health scenarios developed for this dissertation. It was submitted to the Journal of Biomedical Informatics in October 2015, and it is under review.

1.5. Dissertation Structure

This dissertation is organized in eight chapters as presented below.

Chapter 1 – Introduction: it starts to present the main focus of this dissertation. Then, it identifies the research problems about the topic under study, defines the main objectives, and presents the main contributions and the structure of the dissertation.

Chapter 2 - Related Work: it addresses the state-of-the-art, approaching the literature on IoT network protocols and technologies, IPAs, and lastly IPAs inserted on ubiquitous communication scenarios.

Chapter 3 – AMBRO System: it presents the IPA platform where the mobile gateway solution (developed for this project and demonstrated on this dissertation) is inserted. An overview of the system is presented, along with the specification of each architecture layer of the AMBRO platform. There are also specified the components of the architecture.

Chapter 4 – Requirement Analysis: it presents the requirement analysis to obtain all the mobile gateway system necessities, UML diagrams (behavioral, interaction and structural diagrams) and used technologies.

Chapter 5 – Devices and Technologies Used on the Prototype: it indicates all the devices used on the prototype as well as their associated technologies. The specifications of the equipments are also enumerated, including some photographs to show them.

Chapter 6 – AMBRO Mobile Gateway Application: it presents the developed mobile application, presenting the architecture of the AMBRO mobile gateway and the graphical user interface of the application. There are also demonstrated the functionalities of each AMBRO mobile gateway service. To finalize, it is explained how the caretaker can remotely control the services existent on the application.

Chapter 7 – Performance Evaluation, Validation, and Results Analysis: it presents the results returned by the performance evaluation of the system. The conducted experiments evaluate different power consumption strategies for the devices used on the prototype, and different metrics to verify the accuracy of each AMBRO mobile gateway monitoring service. It is also indicated the importance of communication between the mobile gateway and other objects has for the development of an IoT environment.

Chapter 8 – Conclusion and Future Work: it summarizes all the work performed on this dissertation, presenting conclusions, challenges and possible work that can be realized in the future to improve the proposed solution.

2. Related Work

Internet of Things (IoT) is a term that appeared with several definitions, but it is essentially based on an extremely huge network constituted by millions, or billions, of devices that can communicate between them. Several efforts are being made by the scientific community to extend the IoT technology over the world.

Intelligent Personal Assistants (IPAs) are ICT solutions with the ability to help users on their daily activities. The knowledge these devices present about the behavior of their users is very important to improve the assistance they can give. A possible integration of IPAs on IoT environments can improve even more the knowledge of these devices by expanding their knowledge database.

This chapter presents some of the contributions that were already made by other authors about IoT and IPAs integration, presenting some solutions where those two paradigms are combined.

2.1. The Internet of Things

The Internet of Things is seen as one of the potential enablers for ubiquitous communication scenarios [15][16]. It is seen as the Future Internet, where all devices, being mobile or static, battery supplied or not, can communicate within a global network [17]. Even further, the communication is not only made directly between devices, but also between devices and humans [18][19]. To make that happen, some technologic aspects that already exist nowadays can be availed, while others need to be created or improved. This section presents the technologies and the protocols that can enable the creation of IoT networks.

The IoT network can be formed by thousands or millions of heterogeneous devices leading to the use of different protocols and technologies. At this point, neither a standard architecture nor a standard protocol were defined for the IoT [20][21]. Regarding the system architecture, it is considered an initial architecture basing itself on simple contextual aspects. The conceptual architecture is composed by three layers: the perception layer, network layer and application layer, as depicted in Figure 1. The perception layer considers the identification, acquirement, and collection of information by sensing the physical properties of objects using sensors, RFID tags, 1D and 2D bar codes, actuators, and other devices. The main purpose of the network layer is to transmit and process information. The convergence between IoT and industrial technologies occurs in the application layer, enabling real-time applications for different types of industries. The work presented in [22] and [23] highlight this three layered IoT architecture as the initial architecture for the development of IoT projects.

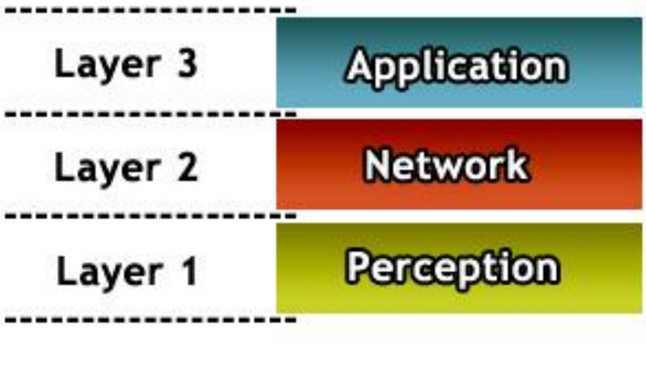

Fig. 1 - IoT conceptual architecture.

In this three-layer IoT architecture, security, reliability, and information processing are not considered. This led to the development of some other, more robust proprietary architectures. Examples include the ISA-100[24], WirelessHART[25], ZigBee Pro[26][27], ZigBee IP[28][29] and Jennic IP[30] architectures. Some of these architectures use the IP protocol (ZigBee IP, for example), while others use proprietary protocols, such as WirelessHART.

To turn IoT into a feasible technology, many standardization efforts have been performed. For the physical and medium access control (MAC) layers, the standard **IEEE 802.15.4** was approved. Created in 2003 by the Institute of Electrical and Electronics Engineering (IEEE), the standard IEEE 802.15.4 aims to be used on low-rate power wireless personal area networks (LR-WPANs) [31]. It has a maximum data rate of 250 kbps, much lower than the rate defined in the IEEE 802.11 standard. This fact allows small and power-constrained nodes to consume less energy. The standard IEEE 802.15.4 has a frame size up to 127 bytes, with two transmission modes, beacon-enabled mode and non-beacon-enabled mode, in the MAC layer. The CSMA-CD is used for collision avoidance in both cases, but in beacon-enabled mode, the device can go from idle to a low-power sleep state to save energy.

Devices using the IEEE 802.15.4 can communicate over areas covering 100 meters on single-hop architectures. Transmissions beyond this range require the cooperation of neighbor nodes to deliver the message to its final destination using a multi-hop architecture. Two types of nodes can exist on a multi-hop architecture: a full-function device (FFD) and a reduced-function device (RFD). An FFD can served as the coordinator of a personal area network (PAN) and communicate with other nodes in the network. RFD has a lower processing capability, so it can never act as the PAN coordinator, communicating only with FFD nodes. There are two network topologies, the star topology and the peer-to-peer topology, as depicted in Figure 2.

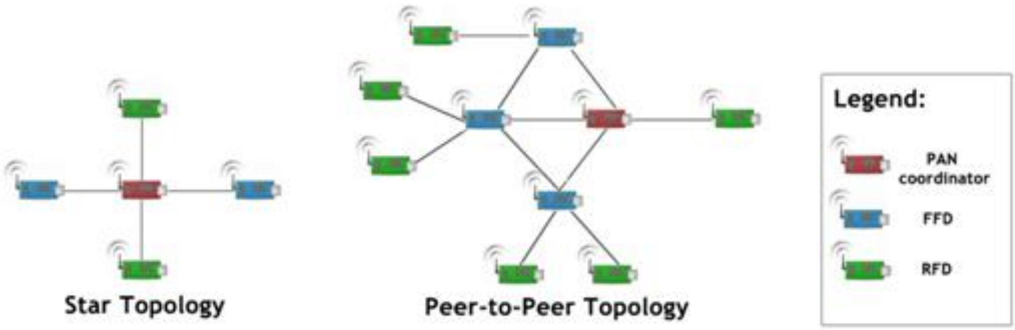

Fig. 2 - Star and peer-to-peer topologies on IEEE 802.15.4.

Ubiquitous communication scenarios can support millions of devices, making the current network addressing scheme insufficient to support all of them. For this reason, it was considered the use of IPv6 as the Internet Protocol in IP-based IoT architectures, since it allows the presence of more devices on the network (2^{128}) compared to IPv4 (up to 2^{32} devices). It also uses the stateless address auto configuration (SLAAC) mechanism to assign an IPv6 address to a device automatically at the moment of its initialization. Using the SLAAC mechanism on WSNs is very advantageous because, on these types of networks, some nodes can be physically inaccessible, making their manual configuration impossible [32]. However, Low Power Wireless Personal Area Networks (LoWPAN) have other characteristics that cannot be fulfilled by the use of IPv6. One example is the impossibility of directly encapsulating IPv6 packets (1,280 bytes MTU) in IEEE 802.15.4 frames. To solve this problem, the **IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN)** protocol was created; it compresses the size of packets flowing in the network, decreasing both the bandwidth and the energy consumed by power-constrained devices. Beyond adapting the data packets' size, 6LoWPAN is capable of resolving addresses, implement addressing management mechanisms, and discover devices and services. All this process translates in the creation of an adaptation layer between the data Link and the network layers of the protocol stack. Figure 3 illustrates the protocol stack of the 6LoWPAN.

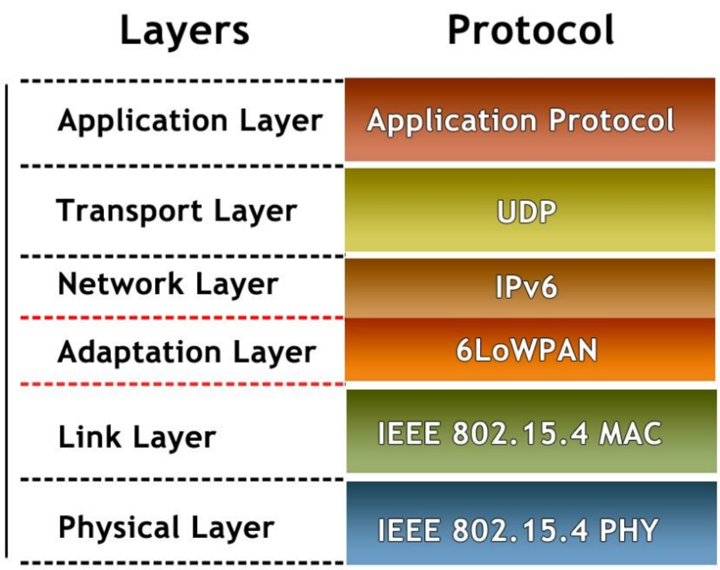

Fig. 3 - 6LoWPAN protocol stack.

Various open-source operating systems are available that implement the 6LoWPAN stack, such as the Contiki OS [33] with its uIPv6 stack, or the TinyOS [34] with its BLIP implementation. Both implementations are important to the deployment of IoT projects using the 6LoWPAN protocol. For example, [35] demonstrated an application that enables an IPv6 connection between IPv6 and 6LoWPAN nodes using the Tiny OS 2.1 BLIP. A gateway is used to connect the 6LoWPAN nodes to the Internet. In [36] is presented a 6LoWPAN implementation over low power and economic CC430-based wireless sensor nodes, giving an insight into the 6LoWPAN protocol and its importance to power constrained devices on IoT deployments.

In 2008, the Internet Engineering Task Force (IETF) Working Group (WG) Routing Over Low-Power and Lossy Networks (ROLL) was created. They intended to create a routing protocol for low power and lossy networks (LLNs) that would support a wide variety of link layers with common requirements such as low bandwidth, lossiness, and low power. That protocol was named **ripple routing protocol (RPL)** [20][37]. The RPL is an IPv6 distance vector routing protocol, oriented for directed acyclic graph (DAG) topologies, in which no direct cycles connect the network devices. Generally, such networks have a root node, or a set of root nodes, to coordinate tasks and collect data. For each of root nodes a destination oriented directed acyclic graph (DODAG) is created using an objective function that defines how the routing metric is computed, and *ranks* to encode the distance of each network node to its reference root. RPL is important to IoT networks by creating multi-hop architectures where power constrained nodes can communicate. Figure 4 shows a DODAG topology with a root node and a set of neighbor nodes. The nodes in the topology have calculated their own distance to the root node of the tree, creating their own ranks.

Various types of messages are used to construct the network graph, called DIO (DODAG Information Object), DIS (DODAG Information Solicitation), and DAO (DODAG Destination Advertisement Object). DIO messages allow a node to discover an RPL graph, learn its configuration parameters, and select DODAG parents. A DAO message is used to spread destination information upward along the DODAG. A DIS message corresponds to the solicitation of a DIO from a RPL node. The RPL also uses mechanisms to repair the network graph when a failure on a link occurs, as well as an adaptive timer mechanism (the trickle timer) to control the rate at which DIO messages are sent to the network [38].

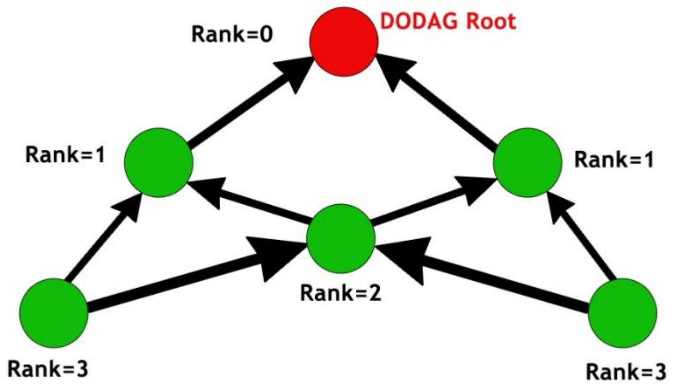

Fig. 4 - DODAG topology illustration.

The RPL can be deployed in various operating systems, such as the Contiki OS and the TinyOS. The authors of [39] addressed the interplay between RPL and various address auto-configuration algorithms, using ContikiRPL to conduct the practical experiments. The work described in [40] evaluated the performance of 6LoWPAN and RPL protocols in TinyOS. The paper gives insight into the characteristics of the RPL standard and its relation with the TinyOS, comparing the TinyRPL solution with the de-facto standard routing protocol for TinyOS 2.X, the collection tree protocol (CTP).

IoT devices can be integrated on the Web, creating the “Web of Things” concept. This fact allows Representational State Transfer (REST) architectures to be used on IoT applications [6][20][41]. Many of those REST architectures use the Hypertext Transfer Protocol (HTTP), which may create performance issues when used on LLNs because of its large overhead. To overcome these issues, the **constrained application protocol (CoAP)** was created in 2010.

CoAP is an application layer protocol based on the REST architecture, where resources are server-controlled abstractions available through an application process and identified by a Universal Resource Identifier (URI). The methods PUT, GET, DELETE and POST used in HTTP are also available on CoAP. Despite using the same methods, CoAP is not a substitute for

HTTP, being easily interoperable through proxy mechanisms. As shown in Figure 5, the main difference is the use of the User Datagram Protocol (UDP) as the transport layer protocol on CoAP and the Transmission Control Protocol (TCP) on the HTTP.

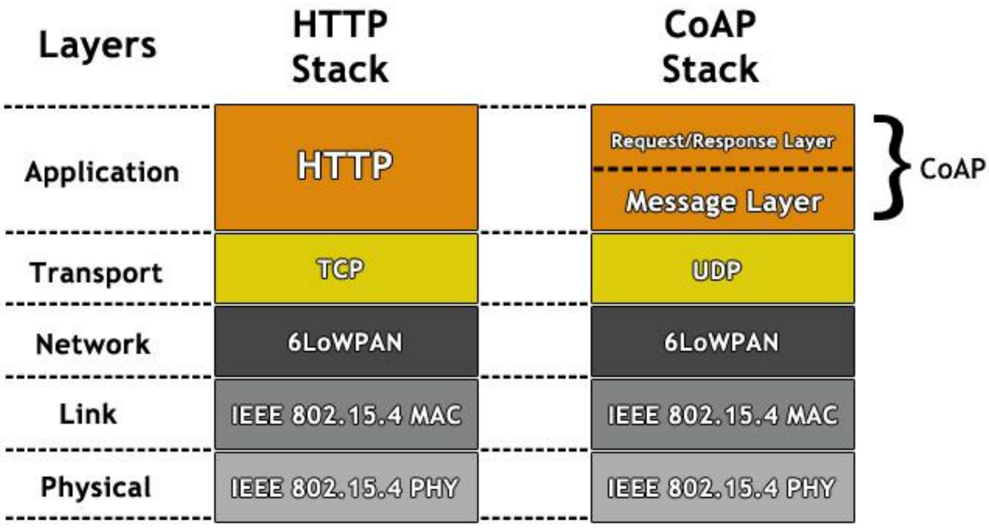

Fig. 5 - Comparison of HTTP and CoAP protocol stacks.

The use of TCP in LLNs is not as appropriated as the use of UDP, because it uses flow control mechanism and reliable transmission methods that, despite being good to assure a trustworthy communication, can make network devices consume more power. Also, TCP has a large overhead (in comparison with UDP) and does not support multicast (TCP is a point-to-point protocol).

The application layer on the CoAP is divided into two sub-layers: message and request/response layer. The message layer controls message exchanges over UDP between two endpoints. Messages are specified by IDs to detect duplicates, assuming one of four different types: confirmable, non-confirmable, acknowledgment and reset. The request/response layer is responsible for the transmission of requests and responses used to manipulate resources. A REST request is piggybacked in a confirmable or non-confirmable message, while a REST response is related to an acknowledgment message [20]. When confirmable messages are used, acknowledgment messages are required to confirm the receipt of the desired information. If a response does not arrive at the client (or the client receives a reset message) the message is retransmitted. With this mechanism, CoAP ensures communication reliability without using TCP. Figure 6 shows an example of a resource request made by a CoAP client to a CoAP server. The client request is made through a confirmable message, with the server answering with an acknowledgment message containing the value of the asked resource.

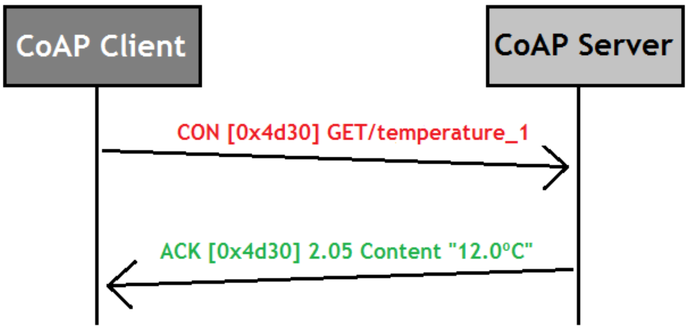

Fig. 6 - Illustration of a CoAP request.

According to the related literature, various CoAP deployments on IoT projects have been implemented. In [42], a comparison is made between CoAP and HTTP through the execution of experiments on client-server scenarios using CoAP on one scenario and HTTP in other scenario. It was concluded that CoAP has a better response time for client requests than HTTP. In [43], the authors presented an IP-based solution to integrate sensor networks properly in a cargo container with logistic processes, using CoAP to obtain the data gathered by sensors (such as temperature or humidity data) during land or sea transportation. The authors of [44] proposed an IoT solution for the mining industry, based on rock bolts and CoAP-enabled sensors. This proposal enables online and real-time monitoring of rock conditions inside mines, detecting potential anomalies at an early stage and raising security levels.

Standardized by the IETF, the **Extensible Message and Presence Protocol (XMPP)**, is an application layer protocol based on the eXtensible Markup Language (XML), which allows two or more entities to communicate inside a network. XMPP is particularly suitable for real-time communication, such as instant messaging. The XML data elements, known as “XML stanzas”, enable the communication inside the network, where, in general, distributed client-server architecture is implemented.

XMPP is a scalable protocol that allows the specification of extension protocols, designated by XMPP extension protocols (XEPs). XEPs allow the addition of new functionalities to the XMPP protocol, introducing the ability to adapt to specific environments. Some XEPs (listed in [45]) can be useful for the IoT domain [46], including XEP-0323, used to provide the architecture, operations, and data structures for communication between sensors on XMPP networks; XEP-0324, used to manage access privileges and to ensure data delivery on existing services; XEP-0325, used to specify the control mechanism on IoT objects; and XEP-0326, which defines how to control architectures that contain servers or concentrators that handle multiple sensors.

The implementation of the XMPP in a IoT scenario is discussed in [47]. The authors presented a lightweight implementation of the XMPP for the Contiki OS, the µXMPP. The µXMPP is configured to follow the LLN specifications, using only essential XMPP characteristics to reduce memory usage on the devices. The authors of [19] showed an improvement to the µXMPP solution with the addition of new modules to the µXMPP architecture, enabling IPv6 support, short JIDs, temporary subscription for presence (TSP) to reduce network traffic, and support for new XEPs, such as XEP-0045 for multi-user chat and XEP-0174 to allow communication without interference of the server. The work described in [48] deployed a service-oriented solution for IoT scenarios, using XMPP as the communication protocol between entities in the network. It considers the diversity of network communication scenarios that exist in the real world and the services that are required by them, such as backend-systems, mobile devices and objects in the environment.

Message Queue Telemetry Protocol (MQTT), developed by the International Business Machines (IBM), is a lightweight application layer protocol based on a publish/subscribe method. With MQTT, clients can subscribe to information (data types) of interest by registering with a broker using TCP connections. Each message data type is referred to as a topic, and clients can subscribe to multiple topics. Every message sent on a topic is received by the clients who have subscribed to it. Figure 7 depicts communication between MQTT publishers and MQTT subscribers using a broker as an intermediary for the communication.

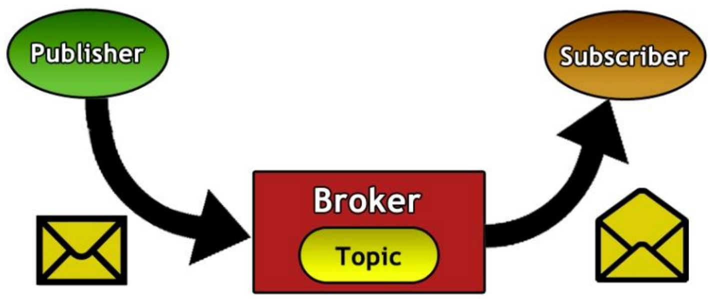

Fig. 7 - Illustration of the MQTT Publish/Subscribe scheme.

MQTT can control the reliability of the end-to-end communication between clients using three different Quality of Service (QoS) levels. QoS level 0, referred as “At most once delivery” is based on a best effort delivery service, on which a message can be delivered to its destination, or not be delivered at all. QoS level 1, or “At least once delivery” guarantees the arrival of messages to the destination by making the emitters send messages to the recipients until they confirm the reception of the original message. QoS level 2, or “Exact once delivery” assures that messages arrive at their destination only once, avoiding duplicate messages. It uses a 4-Way handshake mechanism between the client and the subscriber.

The MQTT protocol is widely used in machine-to-machine (M2M) communication, and it can be used in IoT environments. However, MQTT uses TCP as the transport layer protocol, provoking some inefficiency when used on power-constrained devices. This happens because of the extra power that these devices must consume when data must be retransmitted. To better adapt the MQTT protocol to LoWPANs, MQTT-S (MQTT for sensor networks) was created. MQTT-S defines a UDP mapping for the MQTT, adding broker support for indexing topic names at the same time. The MQTT-S network architecture is constituted by four different elements: a MQTT-S client, a MQTT broker, a MQTT-S gateway and a MQTT-S forwarder. A MQTT-S client connects with the MQTT broker using the MQTT-S protocol. It is used a MQTT-S gateway, which can be incorporated into the broker itself, or implemented in the Low Power Wireless Personal Area Network (LoWPAN) edge router. The gateway makes the correct translation between TCP and UDP, allowing connections between devices on an IPv6 network and an IPv4 network. If the gateway is not directly accessible, a MQTT-S forwarder is used to send messages from the clients to the broker. The MQTT-S architecture can be described as shown in Figure 8 [49]:

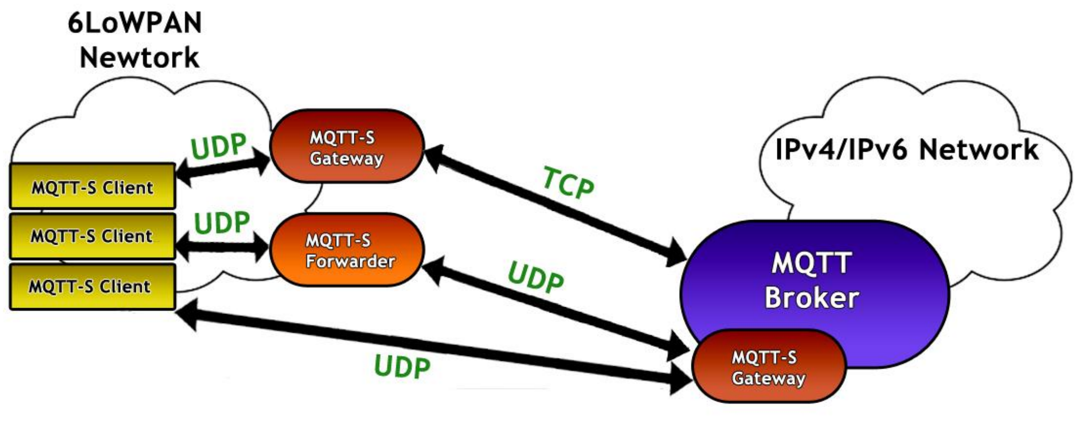

Fig. 8 - Illustration of the MQTT-S publish/subscribe scheme.

MQTT usage in IoT environments was shown in [50], which examined the implementation of the MQTT-S in LLNs. The authors discussed the importance of the publish/subscribe method for WSNs and applications where the content itself is more important than its origin. One of the highlighted MQTT-S characteristics is the possibility of using multiple gateways through a gateway discovery process, which is desirable when link failure scenarios occur. To test the MQTT-S architecture, consisting of a MQTT-S client, a message broker and a MQTT-S gateway, the authors used the IBM wireless sensor networking test bed [51], with two types of clients: ZigBee clients and TinyOS clients.

2.2. Intelligent Personal Assistants

Personal assistants are not recent , having been used in daily business and personal tasks for many years. Personal assistants are responsible for daily task management, such as scheduling meetings, reserving hotel rooms, shopping, and paying bills, among other tasks. In the past, the role of a personal assistant was performed exclusively by people; nowadays, the personal assistant role can be performed by a digital device with learning capabilities.

In the 1990s the IPA concept was introduced. As stated in [52], the use of techniques from the **artificial intelligence (AI)** field created the possibility of constructing intelligent machines that are capable of autonomously performing tasks on the user's behalf. The use of mechanisms such as data mining and machine learning algorithms [53] was important in creating more responsive and self-aware machines. Machine learning algorithms can be divided into two different categories: supervised learning [54] and the unsupervised learning [55] methods. Supervised learning assumes the existence of a labeled training set composed of input-output pairs that make possible the prediction of a new output value from a new input. Supervised learning models include learning decision trees, Bayesian classifiers, linear regression, case-based reasoning (CBR), and neural networks. The unsupervised learning method is more complex, as the training data is not labeled and the machine receives only the inputs. In this model, machines must build representations of the inputs to help in the prediction of new incoming values. The process is based on a search for patterns that can be exploited to create knowledge from input values. Some examples of unsupervised learning models are the K-means clustering algorithm, hidden Markov Models (HMMs) and the *a priori* algorithms.

IPAs can normally interact equally with other intelligent objects in the environment, human or machine, to obtain knowledge about different domains. These scenarios are referred to as **multi-agent systems (MASs)** [56][57]. MASs are composed of multiple heterogeneous interactive intelligent agents within an environment, enabling the parallel processing inside the system and making it less prone to failures. In MASs, agents in the environment can be influenced by other agents or by the environment itself. In single-agent systems (SASs), only the environment can influence the behavior of intelligent agents. In SASs, an agent sees another agent as an integral part of the environment, ignoring the fact that other agents are individual entities in reality. Figure 9 illustrates a scheme that represents the difference between MAS and SAS scenarios.

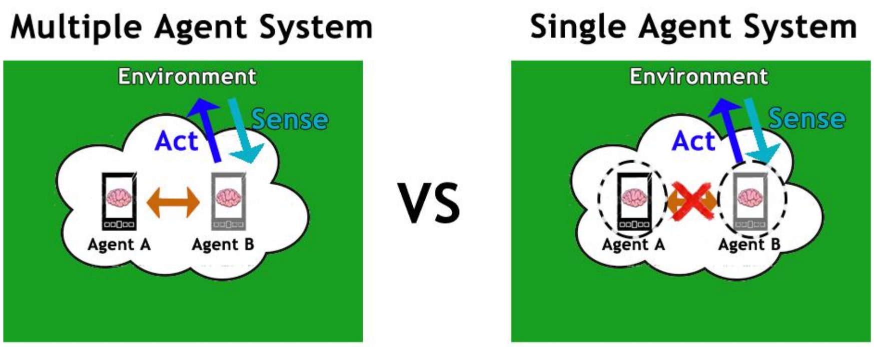

Fig. 9 - Differences between MAS and SAS scenarios.

The way in which intelligent agents in an environment relate among themselves is important in developing complex systems. First, it is necessary to identify the contribution that each individual agent can make to the overall intelligence of the system. This process can be conducted by isolated learning. The learning that agents in the environment perform as a group can be referred to as collective learning. Combining the two types of learning mechanisms and MAS makes possible the design of complex, autonomous, and self aware computational systems.

The technological revolution has enabled the creation of new gadgets with higher computing capabilities and new features, such as smartphones. Some smartphones provide a personal digital assistant as one of their main features. Apple's Siri [58], Google Now [59], Samsung's S Voice [60], LG's Voice Mate [61], and Microsoft's Cortana [62] are examples. Generally, these digital assistants make use of **natural language user interfaces (NLUIs)** to interact with users. NLUIs allow human-machine interaction through the translation of voice commands into machine level commands, an important **speech technology** for human-to-machine (H2M) communication.

The act of making a machine understand human language poses many issues, most of them related with the AI field. In 1952, Bell Laboratories designed a system, known as "Audrey", that recognized digits spoken by a single voice [63]. This experiment focused only on the understanding of numbers since human language understanding was little explored at that time. Other speech recognition experiments were conducted in the 1950s and 1960s, such as IBM's 1962 effort. This system, called "Shoebox", could recognize and respond to 16 words spoken in English.

The 1970s marked the founding of the first commercial speech recognition company, the Threshold Technology, which commercialized the first real automatic speech recognition system, the VIP-100 System. In the same decade, the United States Department of Defense developed a speech recognition system called "Harpy" that could understand up to 1011 words [63].

The 1980s brought great developments in the speech recognition field. A new statistical method to recognize words, HMMs began to replace the templates used to associate words with sound patterns. HMMs are based on the probability of unknown sounds being words, extending the speech recognition vocabulary. However, the human-machine dialogue was not optimal since speech recognition systems required discrete dictation.

The subsequent decades brought great development in the computational processing field with the emergence of computers equipped with faster processors and higher memory capacity. This led to the construction of bigger dictionaries to accommodate human vocabulary known by machines, increasing their knowledge. The processing of words by machines was improved, as they became able to understand continuous speech rather than just isolated words. The development in areas such as automatic speech recognition, natural language understanding, and text-to-speech synthesis were very helpful in constructing today's applications.

The creation of new algorithms to increase the context awareness of machines, support for multiple languages and the development of new mechanisms to recognize specific voices are future steps to improve the speech recognition technology. Figure 10 shows a communication scenario between a user and the Samsung personal assistant, S Voice.

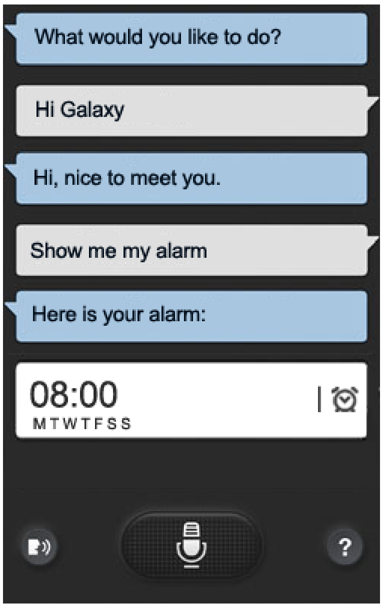

Fig. 10 - Interaction between a user and the S Voice assistant.

H2M communication is a field of great interest in the scientific community, with projects involving IPAs and similar devices existing in great numbers. In [64], the authors discussed existing issues in designing a personal assistant, illustrating some mathematical models to resolve these issues. The authors in [65] is presented an intelligent assistant system composed of intelligent software agents that can help the user with communication, information and time management. Each intelligent agent is responsible for a different task, such as calendar scheduling and telephone call filtering, among others. An intelligent assistant for athletes was proposed in [66], called iAPERAS. It aims to help non-professional athletes who cannot afford their own training team to obtain dedicated information about health indicators. To improve the quality of the information assessment, the iAPERAS system uses Bayesian networks, relying on scientific research findings. The authors of [67] proposed an IPA that allows doctors to monitor the elderly. The IPA, called HealthPal, helps elderly people monitor their health status autonomously, interacting with devices such as thermometers, blood pressure measuring systems, and personal digital assistant (PDA) applications. The medical records are stored on the user's PDA and on the doctor's PC. Notifications are generated when a health problem is detected in the patient. The authors of [68] presented a framework for an intelligent assistant system based on CBR and MAS for manufacturing. The MAS framework allows the creation of a collaborative environment between intelligent agents, decentralizing the processing in the system. Thus, the subsystems throughout the manufacturing system can be seen as individual, intelligent, and collaborative agents. The CBR technique is used to find patterns in the information that is presented based on past experiences. The use of CBR allows the system to find previous orders that are similar to those placed at the time. In [69], the authors presented a pedagogical agent designed to work with Web-based instructional material, called "Adele". This intelligent agent aims to help students learn new subjects by presenting them with some practical cases to solve. The "Adele" agent is also supposed to collaborate with students by making hints, posing questions, and providing explanations. The authors of [70] presented an intelligent assistant for network based communications, discussing the importance of creating a context-aware assistant that can manage communication within the telecommunications network. The proposed agent is based on the Session Initiation Protocol (SIP), which can interact with external information and applications servers. The authors of [71] presented a speech interface for personal assistants used in research and development (R&D) projects. The speech interface presented in the paper is aimed for use in cooperative scenarios with multiple agents. It is used in a knowledge management (KM) multi agent system in which the user can talk in English to the personal assistant to perform various tasks, such as listing all articles available on a certain subject in a database. The personal assistant is supposed to understand what the user says through the use of an ontology handling mechanism that employs the Protégé [72] tool. This paper shows that it is possible to construct a speech recognition interface for IPAs based on projects of different scopes. The authors of [73] proposed a personal assistant to execute services on behalf of users, using natural language interfaces to process the users' requests. For each request, the personal assistant analyzes the phrase and segments it into individual words. Then, it tries to find keywords in the phrase, and a suitable service for the most meaningful keyword is executed. The authors of [74] proposed an IPA that can learn the preferences, goals, and habits of the user by using supervised learning mechanisms, based on the user's previous experiences. The process of monitoring the users is done using Aspect Oriented Programming, breaking down the program logic. There are generated on the user's actions to ascertain whether he/she completed the desired task. If the task is completed, the traces are successful; otherwise the traces are unsuccessful. To enable the IPA to predict the user's actions, the authors used a feed forward neural network, trained with a back propagation algorithm.

2.3. Intelligent Personal Assistants and the Internet of Things

Currently, IPAs are already capable of helping their users perform several tasks. It is necessary to ascertain whether it is possible to improve their knowledge and autonomy by using new technological paradigms. One possible solution would be the insertion of IPAs in ubiquitous communication scenarios. It would enable IPAs to analyze information from a greater variety of sources, increasing the learning database. Such scenarios are possible with the usage of IoT technology.

With the creation of IPAs that are more autonomous and insightful, many tasks performed by users can be eased. Taking, for example a smart home scenario, where an IPA can communicate with other objects in the environment: a person wakes up at 7:00 am and must prepare to go to work. As the alarm clock rings, the curtains in the user's bedroom slowly open to allow some light to enter the room. Three minutes after the alarm clock stops, the person goes to the bathroom to take a shower. After the person takes his shower, the toaster and the coffee machine in the kitchen turn on automatically to prepare breakfast. All these actions are transparent to the user, as it is the IPA that autonomously prepares the user's morning routine. This is possible because the IPA learns the actions that the user normally performs in his daily routine, either by checking the interaction of the user with the IPA itself or through the interaction between the user and other objects in the environment. It is in the cooperation between IPAs and smart objects that the IoT concept is important. Figure 11 depicts the interaction between the IPA, the user, and the smart objects inside a house.

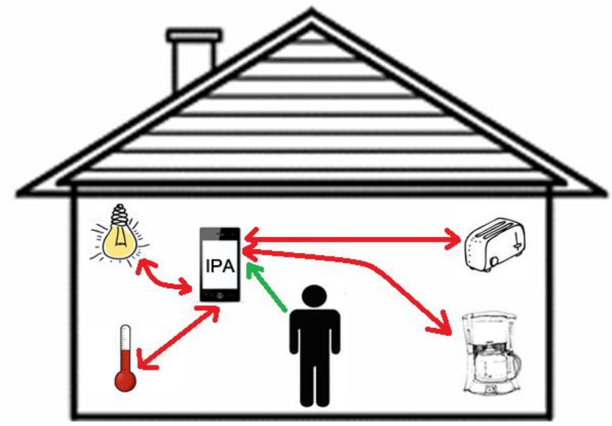

Fig. 11 - Interaction of the IPA and smart objects on behalf of the user.

However, some issues arise from the idea of introducing IPAs in IoT environments, mainly related with technological paradigms. One of these issues is related to the creation of mechanisms to support communication between objects within the network. One possible solution is the creation of a standard IoT communication protocol. Until then, other solutions can be attempted, such as the creation of **network gateways** to map different protocols and networks. The interconnection between IoT environments and external networks has been a subject of research in recent years. In [75], the authors proposed a gateway prototype to connect WSNs with the Internet. The WSN is composed of 6LoWPAN nodes that use short addresses to communicate. The gateway can map these 6LoWPAN addresses into full IPv6 addresses, establishing the connection between the WSN and the Internet. The authors of [76] described the procedures necessary to design a gateway capable of connecting a 6LoWPAN network to an IPv6 network. The solution included the use of low-power sensor nodes configured with the TinyOS 6LoWPAN protocol stack, and an IPv6 client to interact with the same sensor nodes. The gateway was important in compressing the IPv6 header when the communication was done from the IPv6 client to a WSN node and in decompressing the 6LoWPAN header when the communication flowed from the 6LoWPAN WSN. The authors of [77] discussed a gateway that connects 6LoWPAN networks to IPv4/IPv6 networks. The gateway was used to map 6LoWPAN addresses to IPv4\IPv6 addresses and vice versa, building a ubiquitous communication scenario. The authors of [78] described the construction of a prototype for a gateway that connects IEEE 802.15.4 nodes with Ethernet IPv6 networks. The performance of the gateway was evaluated through its capability to compress and decompress IPv6 packets.

With more objects communicating in the same environment, more data are generated. Some of that data may be important to the application scope, making it necessary to store and to manage them. The necessary storage space will be enormous, so new mechanisms must be applied. The use of **cloud computing** technology can be very useful in this case, decentralizing the processing involved in storing and managing the data. The authors of [79] discussed reasons to integrate IoT in the cloud and the difficulties that arise from that integration. The integration of IoT in the cloud led to the development of the concept of CloudIoT, the creation of mechanisms to store data and share computational resources, and a medium where heterogeneous objects can communicate as well. Possible challenges arise from the lack of solutions to manage all the data generated in the IoT network inside the cloud, and the lack of a service to catalog the data generated by all the heterogeneous IoT devices in different categories. The authors of [80] presented a cloud-based IoT architecture, describing a smart home environment containing PCs, PDAs, indoor and outdoor sensors, microwave ovens, toasters, televisions, and stereos, among other appliances. The interaction between those objects leads to some automatic actions that interfere with the daily life of the people who live in the house. The cloud infrastructure is used to store data coming from the objects. The authors of [81] addressed the feasibility of the integration of an IoT network with the Internet using various cloud services. They also described various technologies that can enable the construction of IoT networks, such as WSNs, next-generation networks, and cloud computing platforms. Furthermore, they tested an IoT Cloud platform named Skynet, an instant messaging platform oriented for M2M communication based on the MQTT protocol. The authors of [82] described how IoT and the cloud computing technologies can work together to deal with the huge amount of data that exist in the network. A solution where the sensing of a certain physical measure is considered a service on the cloud was proposed. This allows the creation of a ubiquitous communication scenario where the sensors are the interface between the physical world and the digital world.

In a scenario where an IPA can communicate with every object that surrounds it, extra care must be applied to prevent private information from falling into the wrong hands. These **security** issues are discussed in [16]. The authors identify five IoT security issues, related to (i) the correct identification of IoT objects, (ii) the wireless link insecurities originated by each object connected to the IoT network, (iii) the heterogeneity of network devices and their specific security needs, (iv) the confidentiality of the information flowing on the IoT network, and (v) the information processing security. All these issues can be applicable to scenarios where IPAs are inserted into ubiquitous communication environments, as IPAs can behave as smart objects in an IoT environment. The authors of [83] identified the main problems that can emerge from the creation of IoT networks. Because IoT networks cover a wide range of devices and a great amount of traffic, the authors assumed that it is not possible to use the security mechanisms currently used on the Internet in IoT networks. Problems such as protocol and network security, as well as privacy and data and identity management were enumerated. An IoT scenario where sensors integrated into a smart city environment must send data to a mobile device (like a PDA) was given as an example. In this scenario, new cryptography algorithms, key management systems, and security protocols would be needed to secure the communication. The authors of [84] provided insight into the efforts made by the IETF to develop a standard solution to secure the communication inside an IoT network. The solution uses the CoAP protocol along with Datagram Transport Layer Security (DTLS). Because it is not directly related to existing IPAs, this paper is important in understanding the procedures that must be implemented to introduce IPAs on a secure IPv6 network.

IPAs can be used in many fields for various purposes. For example, the authors of [85] described a solution that uses a wireless body sensor network (WBSN) to monitor the vital signs of multiple patients. The information is passed to the intelligent personal digital assistants (IPDAs) of the medical staff using the ZigBee/IEEE 802.15.4 standard. The IPDA must analyze the data to correctly determine the health status of the patients. The authors of [86] presented a WSN for health monitoring. The system includes wearable devices, whit some of them placed on the body of the patient and others placed around the house. The medical staff is informed about the health status of the patients through notifications received on their PCs and PDAs. The sensors that constitute the network architecture are connected to the backbone (PCs, PDAs, and databases) through motes that use the IEEE 802.15.4 protocol.

Other types of applications that can be enhanced include smart home applications. The AlertMe [87] and the Iris [88] systems are two solutions that can manage home appliances through smartphones. In [89] the authors proposed a remote monitoring and controlling system composed of two sub-systems. One of those sub-systems is used for the real-time monitoring of the house, while the other is used to control the lights in the house, implementing a ZigBee network. The user can see the real-time video footage of the house or control the lights by accessing the Web application or the smartphone application. The authors of [90] described a smart home system composed by an application to control and monitor home appliances, developed for the Android operating system, and an Arduino Ethernet-based micro-Web server. All the sensors in the smart home environment connect with the Arduino platform. The system includes some security features, such as user authentication and the ability to notify the user in the case of any unexpected event (fire in the kitchen, for example). The authors of [91] proposed a novel mechanism for a home appliance control system. The system's user can control the home appliances through a PDA, mobile phone or notebook running the system control application. The home appliances are inserted into a 6LoWPAN WSN, where an appliance control terminal is used to manage each appliance. The system also includes a smart gateway to allow communication between the WSN and other networks.

2.4. Summary

This chapter presented the literature review about the IoT, IPAs, and the possible relationship between both concepts. On Section 2.1, the IoT technology was presented, with the enabling technologies and protocols to deploy IoT networks enumerated. Currently, the main issue on IoT networks deployments relates with the lack of a standard protocol to enable communication between heterogeneous devices. The XMMP, MQTT, and CoAP are some protocols considered by the scientific community to be used as the application protocol for IoT networks.

Section 2.2 presented the IPAs, considering the AI, MAS, and speech technologies as main aspects to create these devices. Section 2.3 presented a review about the insertion of IPAs on ubiquitous communication scenarios. A smart home scenario is given as an example, on which an IPA assisted the user by interacting with other smart objects on the surroundings. A list of possible issues derived from the insertion of IPAs on ubiquitous communication scenarios is also presented. Problems considering the security, data storage, and communication between IoT and external networks are highlighted

3. AMBRO System

This chapter presents the AMBRO system, a platform that allows the integration of IPAs on ubiquitous communication environments. The mobile gateway proposed on this dissertation is one of the components of the AMBRO system. The AMBRO system architecture and its main components are described along this chapter, emphasizing the main functionalities of the system.

3.1. System Overview

AMBRO is a technological platform that allows the integration of IPAs on ubiquitous communication environments. AMBRO was developed to assist his user on innumerous ways by giving him hints and suggestions about several aspects of his daily life, while interacting with several entities in the environment. By interacting with AMBRO, an user can: (i) access Web services to consult aspects of his interest (find nearer restaurants, consult the weather, among others), (ii) examine information of sensors on his possession (see temperature at home, for example), and (iii) change the behavior of smart objects in the environment (turn off the air conditioner, among others). AMBRO can autonomously interact with his user to give him advices about subjects of his interest because it can learn the user behavior over the time. This learning occurs through successive direct interactions with the user, or by understanding the user interactions with other objects in the environment. Figure 12 shows an example of a scenario where the user consults relevant information about his objects (image on the left), and a scenario on which the user interacts with the AMBRO personal assistant to turn on the toaster (image on the right).

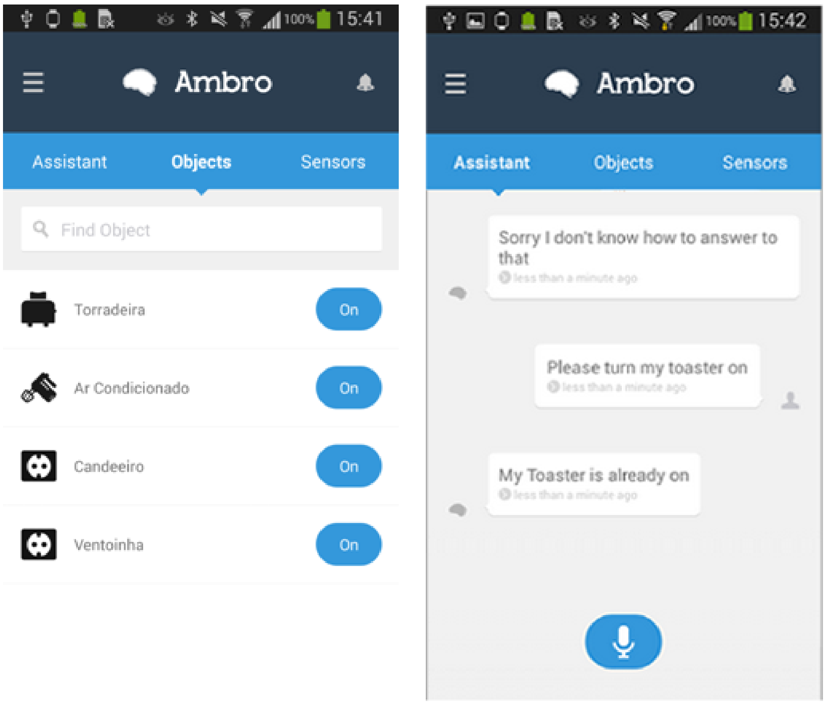

Fig. 12 - Example of AMBRO IPA menus.

3.2. AMBRO System Architecture

The AMBRO system architecture is composed by three layers. The distribution of the system functionalities on various layers eases the detection and correction of problems that might appear during the system operation. The three main layers are [92]:

- The **IoT World** environment, on which all the external devices that give information to the AMBRO's personal assistant about the environment that surrounds him are present;

- The **Intelligent Cloud System** environment, which is the core of the system, and allows the communication with other modules, management of the data, and treatment of business logic;
- The **User Interfaces** environment which hosts the AMBRO interfaces. Users can use the AMBRO interfaces to interact with the personal assistant.

Figure 13 illustrates the AMBRO system architecture.

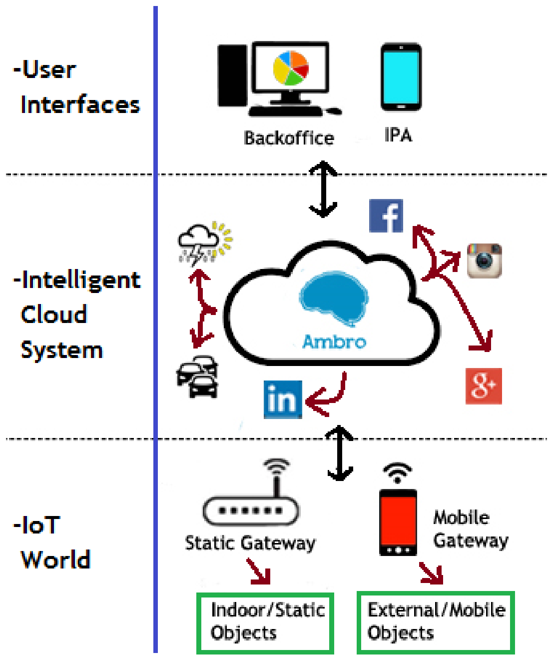

Fig. 13 - Illustration of the AMBRO system architecture.

The AMBRO architecture can also be divided on a logical perspective. It gives a detailed insight about the system architecture, showing the modules and respective specifications for each one. The logical units of the AMBRO architecture are:

- **Channels**, presenting the IPAs interfaces available for the user;
- **Interface**, representing the communication channel between the AMBRO services and the IoT elements;

- **Ambro**, which groups the logical operations behind the AMBRO framework, and allows the integration of all the subunits of the central processing unit;
- **IA**, which combines all the speech recognition and artificial intelligence algorithms;
- **Comm**, which assures the communication between AMBRO and external entities;
- **Events**, which allows the registration and treatment of data coming from IoT objects;
- **IA tools**, which is used by IPAs to learn the usual behavior of the user and the usual behavior of the IoT objects;
- **WSO2**, which is responsible for managing the tokens and user sessions. It also filters and validates the access to the AMBRO cloud;
- **DB**, presenting the database of the system. Maria DB is used to save data concerning the users, objects, and system events.

Figure 14 depicts the logical architecture of the system.

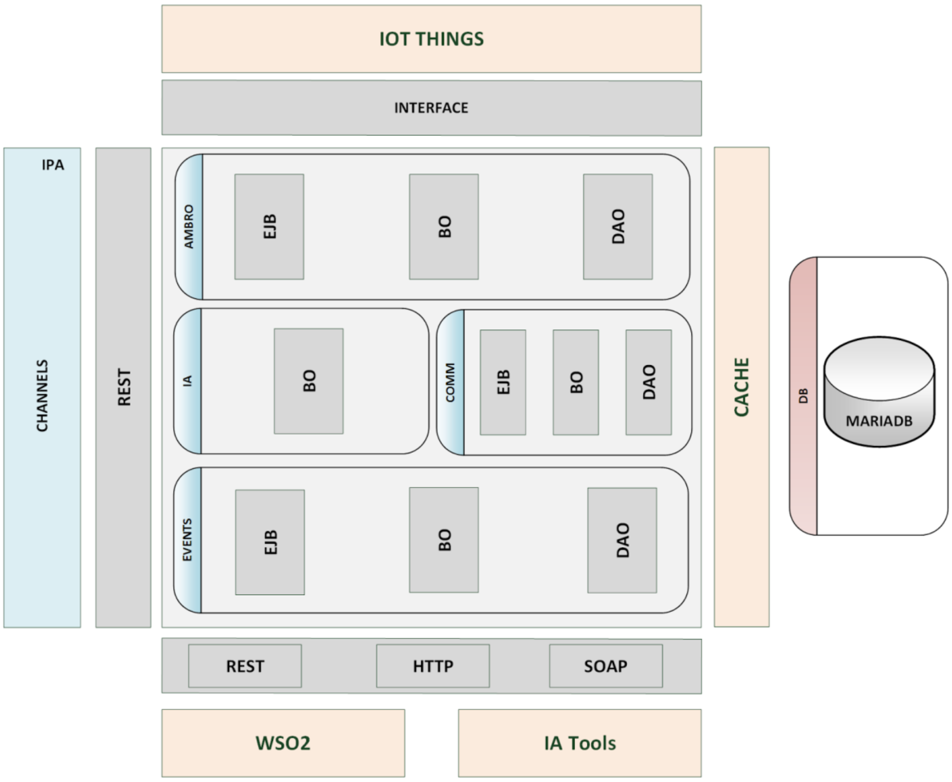

Fig. 14 - AMBRO logical architecture.

3.2.1. AMBRO Components

The three main layers of the AMBRO architecture can be deeply analyzed and create a set of modules that can better explain the system functionalities. The **IoT World** environment comprises three main components: gateways, sensors and objects. Figure 15 depicts the IoT World layer.

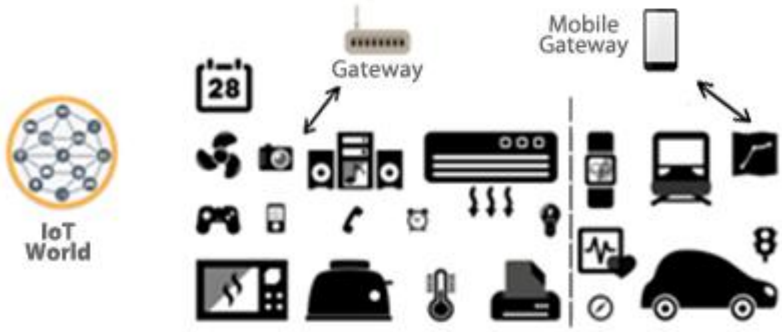

Fig. 15 - IoT World layer illustration.

The static gateway on the IoT World layer is responsible for the communication management between sensors and objects in the environment, and the cloud system. The gateway integrates different protocols and communication technologies such as ZigBee and Z-Wave [93]. An AMBRO proprietary protocol is also supported by the gateway. The gateway was developed using the NodeJs tool and has two main functionalities: (i) discover devices in the environment, and (ii) mediate the message exchange between devices in the environment and the cloud server. The static gateway uses different techniques and methodologies to connect with the surrounding environment. For scenarios where AMBRO sensors and AMBRO objects are considered, the gateway operates as a socket server while sensors and objects are socket clients. For ZigBee and Z-Wave devices, the respective connection techniques mentioned by each brand are used. To connect with the cloud server, the gateway behaves as a socket client.

The mobile gateway is used to connect sensors monitoring a person on a WBSN scenario to the AMBRO platform. It is developed for the Android operating system, and connects with AMBRO cloud using RESTful Web services. It also allows the discovery of Bluetooth devices usable on services existent on the mobile gateway.

Sensors are used to retrieve information on the user environment. The periodicity used by sensors to gather information about the environment variables is carried out independently, which allows the user to adjust it as he wants it. Objects as fans or light bulbs can be operated automatically or through the user action.

The **Intelligent Cloud System** unifies the transactions made between system components. It implements the AI mechanisms and messages/ notifications routing from AMBRO entities. Figure 16 shows an illustration of the Intelligent Cloud System layer.

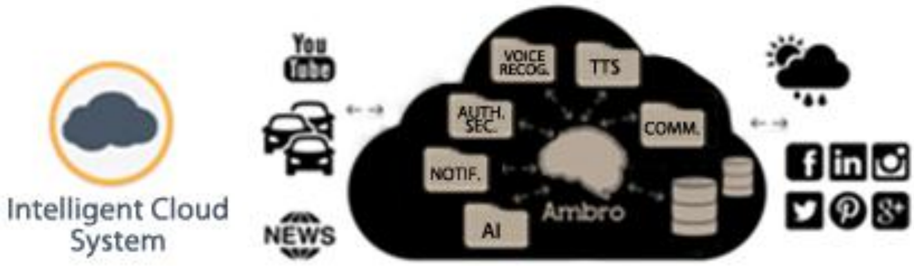

Fig. 16 - Illustration of the Intelligent Cloud System layer.

The Authentication and Security module is used to authenticate all the users that want to access the AMBRO system. The WSO2 Identity Server (IS) and the WSO2 Enterprise Server Bus (ESB) frameworks were used. The WSO2 IS is used to manage the security and identity for applications, services and APIs. The WSO2 ESB provides secure access to system endpoints. Tokens used to provide system access to the users are generated at the WSO2 ESB.

The Artificial Intelligence (AI) module allows learning machine algorithms to be applied on the system. Training, classification and learning methods are used on the information made available for the system (like user behaviors). It allows IPAs to learn the behavior of their users in order to improve the assistance they give.

The Text-to-speech (TTS) module synthesizes text strings and files into spoken audio using synthetic voices. The Voice recognition module converts human spoken audio into readable text strings and files. Both TTS and Voice recognition modules use the Microsoft Speech API (SAPI). Those two modules combined originate an intelligent speech module able to comprehend requests from the users and give suitable answers to those same requests.

The existence of the Notifications module is important to allow notifications about events on the environment to be sent to the IPAs. Notifications can behave as reminders to inform the user about some event that is about to happen (or has just happened), or as an alarm to notify the user about a critical and unexpected event on his environment.

The Communication module is responsible for the communication between AMBRO entities and external devices. The interaction between IPAs and the Web services (such as Google Places, news services, weather services, among others), and the interaction between IPAs and the gateways are managed on this module.

The management and storage of all the data generated on the system are also made on the Intelligent Cloud System layer. For that purpose is used the Maria DB.

The **User Interfaces** layer comprises the back office tool and all the different IPA interfaces that allow users to interact with AMBRO. Figure 17 illustrates the User Interfaces layer of the AMBRO system.

Fig. 17 - Illustration of the User Interfaces layer.

The back office tool is important for the detailed management of the AMBRO environment. It is composed by two subsections, one of them oriented for the service provider and another oriented for the user. The subsection oriented for the service provider is intended for the management of multiple AMBRO accounts corresponding to different clients. This subsection is also responsible for the association of gateways to the clients. The subsection directed to the user has a set of features equivalent to the ones presented by IPAs. However, it also adds management and administrative capabilities to the personal assistant. It has an additional set of tools that allows the monitoring of multiple features on the communication environment as well as an area to schedule tasks and automate procedures.

The IPA interfaces layer gives users the possibility to interact with the assistant and its surrounding environment through direct commands, and remote commands to interact with objects and sensors. Also, each IPA has the ability to notify the user about significant events and actions important for him. The AMBRO personal assistant was developed for three platforms: (i) mobile, (ii) Web and (iii) smartwatch. The IPA mobile interface was developed for two operating systems, Android (minimum version 4.0.3) and iOS (minimum version 8.1). The IPA Web was developed to allow a user to access AMBRO on every Internet enabled device. The smartwatch interface was created with the purpose of exploring a relative recent trend, the wearable device technology. The IPA smartwatch does not present all the functionalities available on the IPA mobile and IPA Web interfaces, because its interface is reduced. Despite that, the IPA smartwatch interface allows users to interact with the personal assistant, objects and sensors. Notifications are also received on the IPA smartwatch. The Android Wear operating system (minimum version 5.0.2) was used to develop the IPA smartwatch interface.

3.3. Summary

On chapter 3 was presented an overview of the AMBRO system. The AMBRO main purpose was described on Section 3.1. At Section 3.2 was presented the AMBRO architecture, including the description of the three main layers. The IoT World environment includes all objects, sensors and gateways of the system. The Intelligent Cloud System is the brain of the system, including modules to manage the communication, authentication of users, and security of the system. This module is also responsible to make IPAs learn the behavior of the user through the use of learning machines algorithms. Other features, such as speech recognition and notification generation are also present on the Intelligent Cloud System module. The User Interfaces layer, which is the last layer of the system, includes all the interfaces that user may use to interact with the AMBRO personal assistant. A back office tool to manage various features of the AMBRO environment is also present on the User Interfaces layer. Lastly, this section summarizes the chapter 3.

4. Requirements Analysis

Through the analysis of the system requirements is possible to identify the characteristics and behaviours the system can include and support. The requirements are the functional specifications of the system, being important for the system management.

The Unified Modelling Language (UML) can be used to set the system requirements. UML diagrams allow system architectures to be visualized, defining the system activities, components, and behaviours, among other components. UML is not a development method by itself, but it is compatible with other object-oriented software development methods, as the Object modelling technique (OMT). Using UML diagrams enable system models to be viewed on two perspectives: structural and behavioural. Structural diagrams give insight about the system static structure, using objects, operations, and relationships, among others. Behavioural diagrams are used at the analysis of the system dynamic behaviour, pointing the collaboration between objects and their internal state change.

This chapter will focus on the specification of the requirements for the AMBRO mobile gateway system.

4.1. Essential Modelling Requirements

Determining the essential modelling requirements is important for the development of the desired application. Thus, it is possible to predict the application behavior and avoid possible problems arising from a poor design planning. The system features and constraints need to be identified at early development stages.

The essential requirements defined for the AMBRO mobile gateway application were:

- Android API level have to be equal or above the level 18 (Android OS version 4.3) to allow all the services to work correctly. This is due to the usage of the Bluetooth Low Energy (BLE) technology on the connection between the smartphone and the smartwatch used on the heart rate monitoring service;
- All devices used in the project shall be Bluetooth enabled devices;
- The device performing the mobile gateway role must support Wi-Fi and other data networks (3G, 3.5G, 4G);
- The monitoring services available on the mobile gateway have to work on the background without interfering with other activities performed by the user;
- The mobile gateway must always be connected to the Internet;
- The mobile gateway should assure the delivering of all the information retrieved by the monitoring services to the AMBRO platform;
- The mobile gateway should assure the remote control of the monitoring services by the caretaker.

4.2. Behavioral Diagrams

Behavioral diagrams explain some of the system functionalities by using activity diagrams and sequence diagrams. Actions performed by certain entities on the system are described by use case diagrams. Table I shows the main actions performed by the AMBRO mobile gateway system entities.

Table I - Actions performed by each system actor.

Actors	Process characteristics
Monitored Person	• Login; • Logout; • Start and stop the monitoring services; • See the coordinates of his current location; • See his actual heartbeat; • See the number of falls detected by the Fall Detection Service; • Receive notifications about the remote activation/deactivation of the monitoring services.
Caretaker	• Enable/disable services through the AMBRO back office; • Receive notifications about unusual values detected by the monitoring services.

Mobile gateway	• Establish a Bluetooth connection with the smartwatch; • Establish a Bluetooth connection with Shimmer modules; • Send Bluetooth broadcast messages to the surrounding environment to find Bluetooth objects every time the user requests it; • Enable the GPS embedded in the smartphone every time it is required; • Periodically send the data gathered from the Location Monitor Service to the AMBRO cloud; • Disable the GPS location updates when the user is not moving; • Periodically send the data gathered from the Heart Rate Monitor Service to the AMBRO cloud; • Send a notification to the AMBRO cloud every time a fall has been detected.
GPS sensor	• Embedded in the mobile gateway, it retrieves the current user location coordinates.
Smartwatch **(Optical heart rate sensor)**	• At a one second rate, it checks the current user heart rate through a embedded optical heart rate sensor; • Through a Bluetooth connection, it sends the heart rate values of the user to the mobile gateway; • Calculate the mean of the heart rate of the user at a five minutes rate and sends it to the mobile gateway.
Shimmer module	• Establish a Bluetooth connection with the mobile gateway; • Detect falls by measuring the acceleration and the angular velocity periodically.

As described on Table I, there were identified six main system entities: (i) the monitored person, which is the person directly using the AMBRO mobile gateway, (ii) the caretaker, which possesses the IPA personal assistant and is responsible for taking care of the monitored person, (iii) the mobile gateway, that connects the BSN sensors with the AMBRO system, (iv) the GPS sensor used on the location monitoring service, (v) the smartwatch, using an embedded optical heart rate sensor for the operation of the heart rate monitoring service, and (vi) the Shimmer module, which uses an accelerometer and a gyroscope to detect possible falls while the fall detection service is running.

4.2.1. Use Case Diagrams

Use case diagrams show actions performed by the entities of the system. The actions are defined as use cases, since they describe the interaction between the system entities and the system itself. Figure 18 shows the use case diagram for the monitored person.

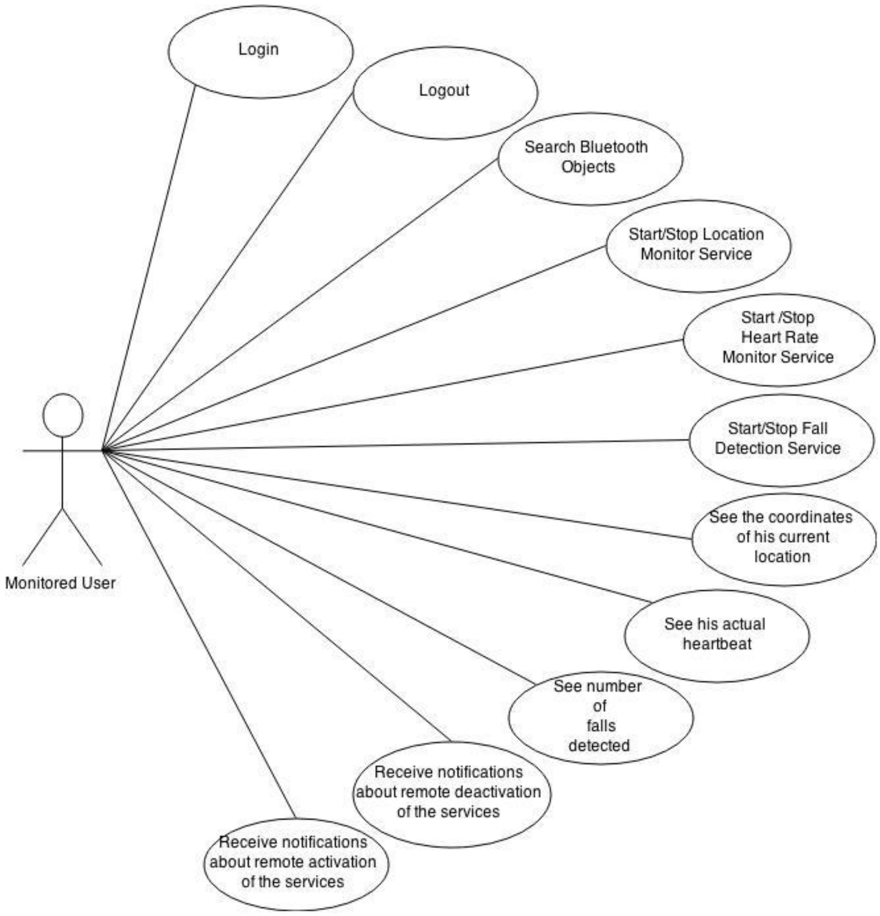

Fig. 18 - Use case diagram of the monitored user.

The caretaker can also interact with the AMBRO mobile gateway application. Figure 19 shows the use case diagram illustrating the interaction between the caretaker and the AMBRO mobile gateway.

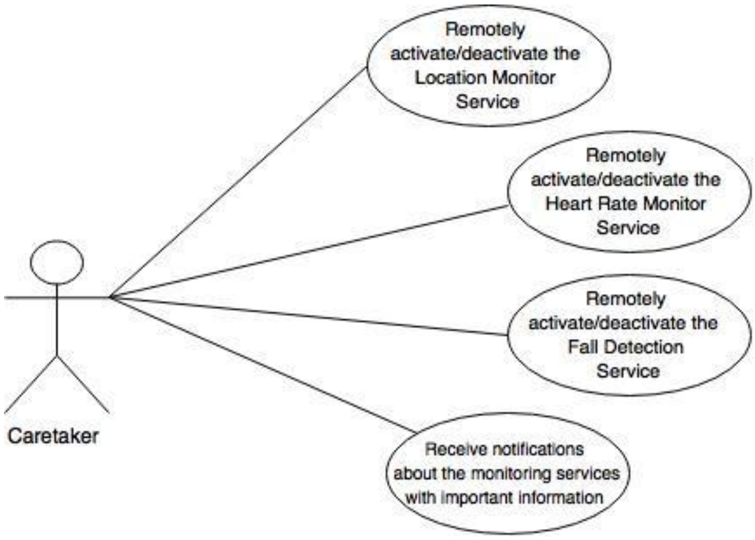

Fig. 19 - Use case diagram of the caretaker.

4.2.2. Activity Diagrams

Activity diagrams are used to describe the workflow of each system activity. On the AMBRO mobile gateway system the activity diagrams explain every step involved on the operation of each monitoring service. An activity diagram is designed to explain the methodology used on the location monitoring service to detect the user activity and location. Another activity diagram explains how the heart rate is detected on the heart rate monitoring service. The third diagram is used to explain the conditions on which the accelerometer and the gyroscope of the Shimmer module are used to detect a possible fall.

4.2.2.1. Location Monitoring Service Activity Diagram

The location monitoring service uses the smartphone GPS system to accurately retrieve the location of the AMBRO mobile gateway user. The service is able to detect if the user is inside a building or at an open space, choosing the best location provider available. For enclosed spaces are generally used the network or power cell location providers. For open spaces is used the GPS system, giving more accurate results. Besides that, the location monitoring service is also able to check the activity the user is performing using the low power embedded sensors of the smartphone (accelerometer, gyroscope, barometer, among others).

Other works have been published concerning tracking systems similar to the one being presented on the AMBRO mobile gateway. For example, the authors of [94] presented a location based application for Android, capable of changing the profile on the smartphone accordingly to the user location; locating a person through a SMS; and locating people on nearby places to the user. The authors of [95] proposed a vehicle tracking system using the GPS system of an Android smartphone. The application can be useful to monitor the driver performance, check the speed at which a vehicle circulates, or increase the security and performance on the fleet management businesses. The authors of [96] proposed a smartphone tracking system to retrieve children’s current location. The application is designed to help parents track their children in real time by sending SMS messages to the children smartphones. The children’s smartphones answer with their current location signaled on a map. On [97], the authors proposed a mobile tracking application using the GPS system of an Android smartphone. The application has the purpose of tracking children through their smartphones, defining geographical areas that cannot be surpassed. The parents are warned through a SMS message if their child is outside the geographical area allowed.

The AMBRO mobile gateway application is used on a ubiquitous mobile heath scenario. The application is used to give the current location of a monitored person to the caretaker. The caretaker (using the AMBRO IPA) is always warned if the person under his care surpasses the defined geofences. These geofences are generated dynamically on the Artificial Intelligence module of the AMBRO cloud (which is described on Chapter 3), since the system is able to learn the usual location of the person being monitored. Figure 20 presents the activity diagram for the location monitoring service of the AMBRO mobile gateway.

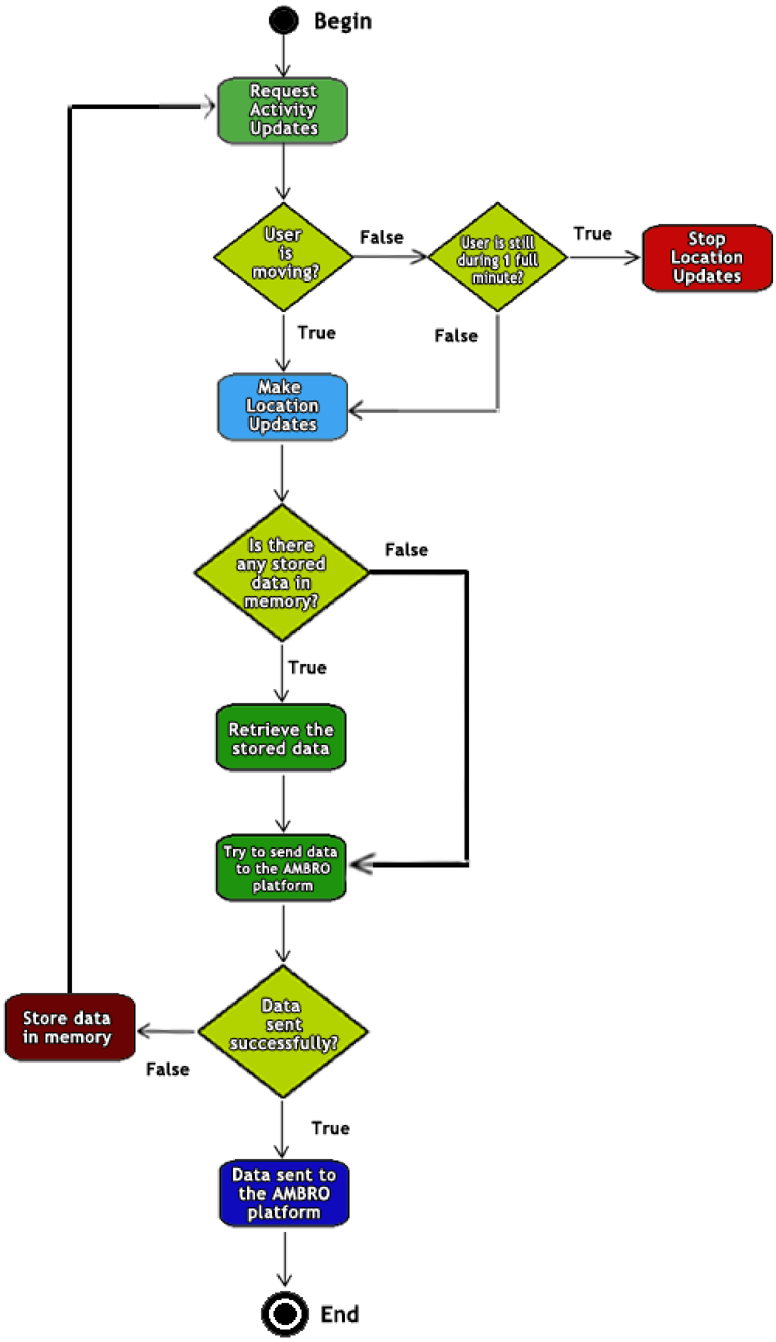

Fig. 20 - Location monitoring service activity diagram.

When the location monitoring service is initiated, activity updates are requested from 20 to 20 seconds. GPS is turned on and location updates are requested every time the smartphone detects the user is moving. Location updates are stopped if the smartphone detects the user is still three times in a row (1 minute still). Regardless of the activity detected, location updates are always triggered with a periodicity of an hour.

The location monitoring service allows data (coordinates and activity detected) to be stored on the smartphone memory if the Internet connection fails. The smartphone retries to send the data to the cloud every time an activity detection update is requested. This mechanism assures that no location updates are lost.

4.2.2.2. Heart Rate Monitoring Service Activity Diagram

In order to make the heart rate monitoring service works, it is necessary to establish a BLE connection between the mobile gateway and the smartwatch. When the connection is established, the optical heart rate sensor embedded on the smartwatch can be used to retrieve the heart rate of the user. Figure 21 shows the heart rate monitoring service activity diagram.

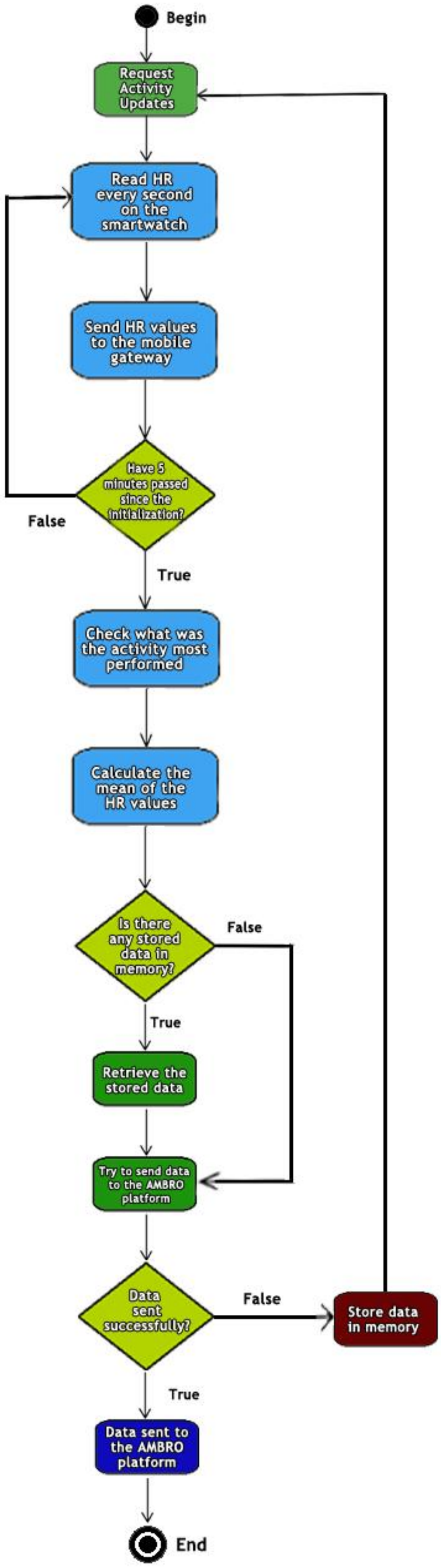

Fig. 21 - Heart rate monitoring service activity diagram.

When the service is initialized activity updates are requested. The low power embedded sensors on the smartphone are used to retrieve the activity performed by the user with a 20 seconds periodicity.

The user heartbeat count is checked every second on the smartwatch. The heart rate value detected is sent to the AMBRO mobile gateway through a BLE connection. The heart rate values detected on the smartwatch are summed up on the gateway, and a mean of the user heart rate on the past five minutes is calculated. Every five minutes the gateway tries to send to the AMBRO cloud the information regarding the activity most performed by the user and his average heart rate on the last five minutes.

On the AMBRO cloud there are two different typical heartbeat mean values for the user: one value is used for scenarios where the user is still; and another value for scenarios where the user is in movement. Those values are obtained through the use of AI algorithms. A notification is sent to the IPA of the caretaker every time a heartbeat value coming to the AMBRO platform is too high or too low when compared with the typical heartbeat mean values of the user. Therefore, the caretaker is informed of possible health problems on the person under his care by evaluating odd heartbeat values detected by the heart rate monitoring service.

If the data is not successfully delivered to the cloud, the mobile gateway stores the information on its memory to send it later. This process is performed every 5 minutes.

4.2.2.3. Fall Detection Service Activity Diagram

A fall can be detected by using the values gathered by an accelerometer and a gyroscope. The total sum of the acceleration on the three axis of an accelerometer can be used to detect a fall by using the equation (1).

$$|A| = \frac{\sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2}}{9.8} \qquad (1)$$

The acceleration value obtained from the calculation is compared with pre-defined thresholds to see if a fall is detected. When a person is still, the acceleration measured in relation to the gravity is approximately 1 G. When a person is at free fall the acceleration value approaches zero G's, and when a person hits the floor that value rises, reaching in many situations, values higher than 2.5 G. The use of a gyroscope can improve the performance of fall detection systems. Gyroscopes are used to detect the orientation of a person by calculating the angular velocity using the equation (2). Generally, when a person falls, the value of the angular velocity is higher than 200º.

$$|\omega| = \sqrt{(\omega_x)^2 + (\omega_y)^2 + (\omega_z)^2} \qquad (2)$$

In the scientific community there exists some work published about fall detection systems. The authors of [98] tested a fall detection system using a tri-axial accelerometer and a tri-axis gyroscope. By collecting some data through experiments performed during various daily activities, the authors defined upper and lower thresholds for both acceleration and angular velocity. By using these thresholds, falls could be detected through acceleration and angular velocity values measured in real-time. Authors of [99], used a 3-axis accelerometer and magnetometer of a smartphone to detect falls. Thresholds were defined for both accelerometer and magnetometer data, defining an algorithm able to detect falls when certain values below lower thresholds and values above upper thresholds were obtained. In [100], the authors demonstrated a real-time wireless fall detection system using a 3-axis accelerometer, capable of differentiating falls from typical activities performed on the daily life routine.

In order to correctly detect a fall on a fall detection system, as the one implemented on the AMBRO mobile gateway application, thresholds need to be defined for acceleration and angular velocity values. To define those thresholds, experiments were made with the Shimmer module while a person was performing some daily life activities. Table II shows the values for both acceleration and angular velocity measured by the Shimmer module when tested on different life activities.

Table II - Acceleration and Angular Velocity values for different daily life activities.

Activities Performed	Acceleration (estimate) (g)	Angular Velocity (estimate) (degrees/sec)
Still	0.9 - 1.15	0 - 90
Fall	0.05 - 3.5	150 - 300
Walk	0.4 - 1.7	0 - 100
Sit down	0.3 - 2.4	30 - 140
Stand up	0.6 - 1.4	30 - 140
Lay down	0.5 - 1.5	80 - 200
Run	0.2 - 2.9	15 - 170
Jump	0.4 - 3	80 - 170

As the Table II shows, when a fall is detected the acceleration varies from 0.05 G to 3.5 G. The lower acceleration value is detected when a person is at free fall. The higher acceleration value is usually correspondent to the instant the person hits the floor. Figure 22 illustrates a situation where a person falls. The values for both acceleration and angular velocity were gathered using the Shimmer sensor module.

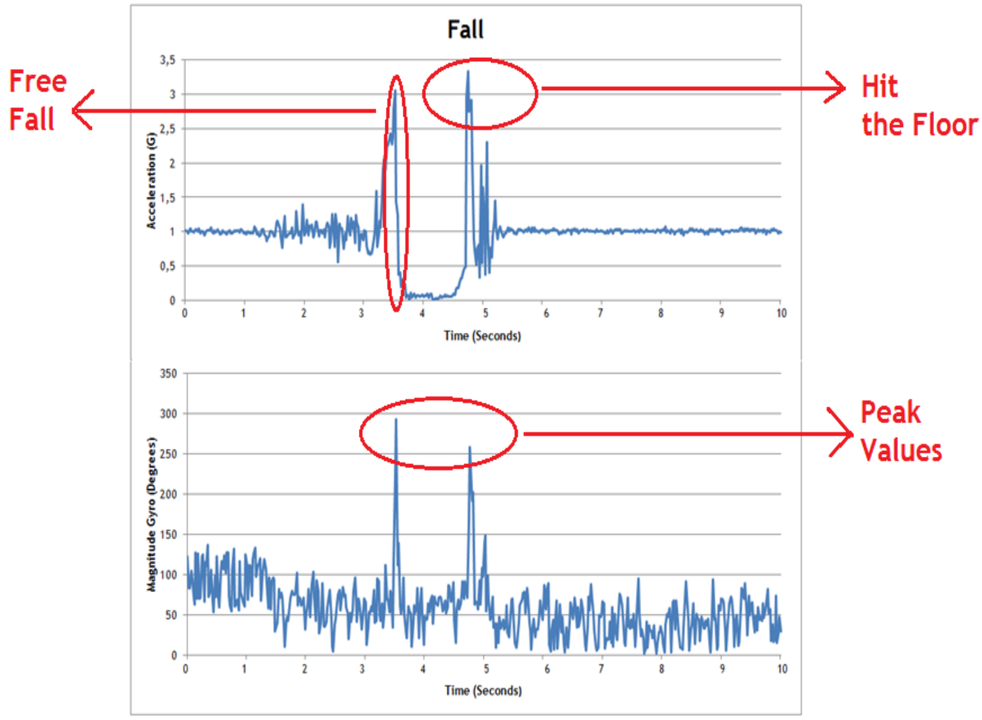

Fig. 22 - Graphics showing acceleration and the angular velocity values that were measured during a fall.

Thresholds for acceleration and for angular velocity can be defined through the analysis of the Table 2 and Figure 16. The Lower Acceleration Threshold (LAT) was defined as 0.3 G while Upper Acceleration Threshold (UAT) was defined as 2.5 G. The Angular Velocity Threshold (AVT) was defined for the value of 220º.

Figure 23 shows the activity diagram for the fall detection service.

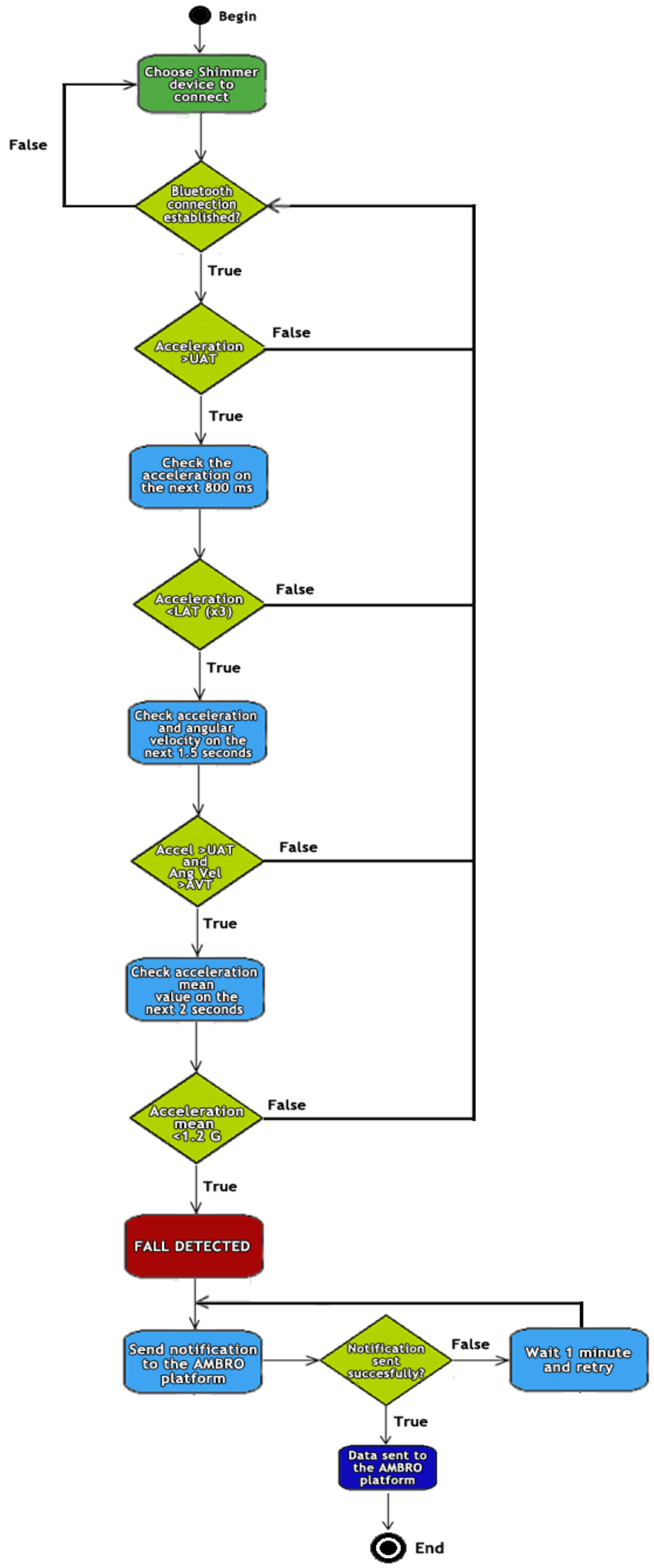

Fig. 23 - Fall detection service activity diagram.

First, the user chooses the Shimmer sensor that he wants to use on the fall detection service. After that, the mobile gateway has to establish a Bluetooth connection with the Shimmer sensor module that was chosen. The Shimmer module starts streaming every 20 milliseconds, gathering values from the accelerometer and the gyroscope. A fall is detected every time the UAT is exceeded, and in the next 0.8 seconds are detected at least three values below the LAT. If this premise is fulfilled, it is possible for a person to be at a free fall. To confirm that a person is effectively falling, it is necessary to ascertain if in the next 1.5 seconds the acceleration surpasses again the UAT and the angular velocity exceeds the AVT. If all this these premises are true, one last step must be performed: the average value for the acceleration in the next two seconds needs to be less than 1.2 G, indicating that the person is still, and possibly lying on the floor. This means a fall has been detected, and the AMBRO mobile gateway will send a notification to the IPA of the caretaker to inform him about the situation. If the notification fails to be delivered, the mobile gateway will try to resend it one minute later. This process will repeat until the notification has been delivered or the fall detection service has been stopped by the user.

4.2.3. Sequence Diagrams

Sequence diagrams depict the action flow between system entities, identifying their behavior. The sequence diagrams presented define the interaction between (i) the monitored person, the mobile gateway, and the sensors for the monitoring services scenario, and (ii) the interaction between the AMBRO mobile gateway, the AMBRO cloud, and the IPA of the caretaker to illustrate the relationship between the AMBRO mobile gateway and the AMBRO personal assistant.

Figure 24 shows the sequence diagram of the location monitoring service.

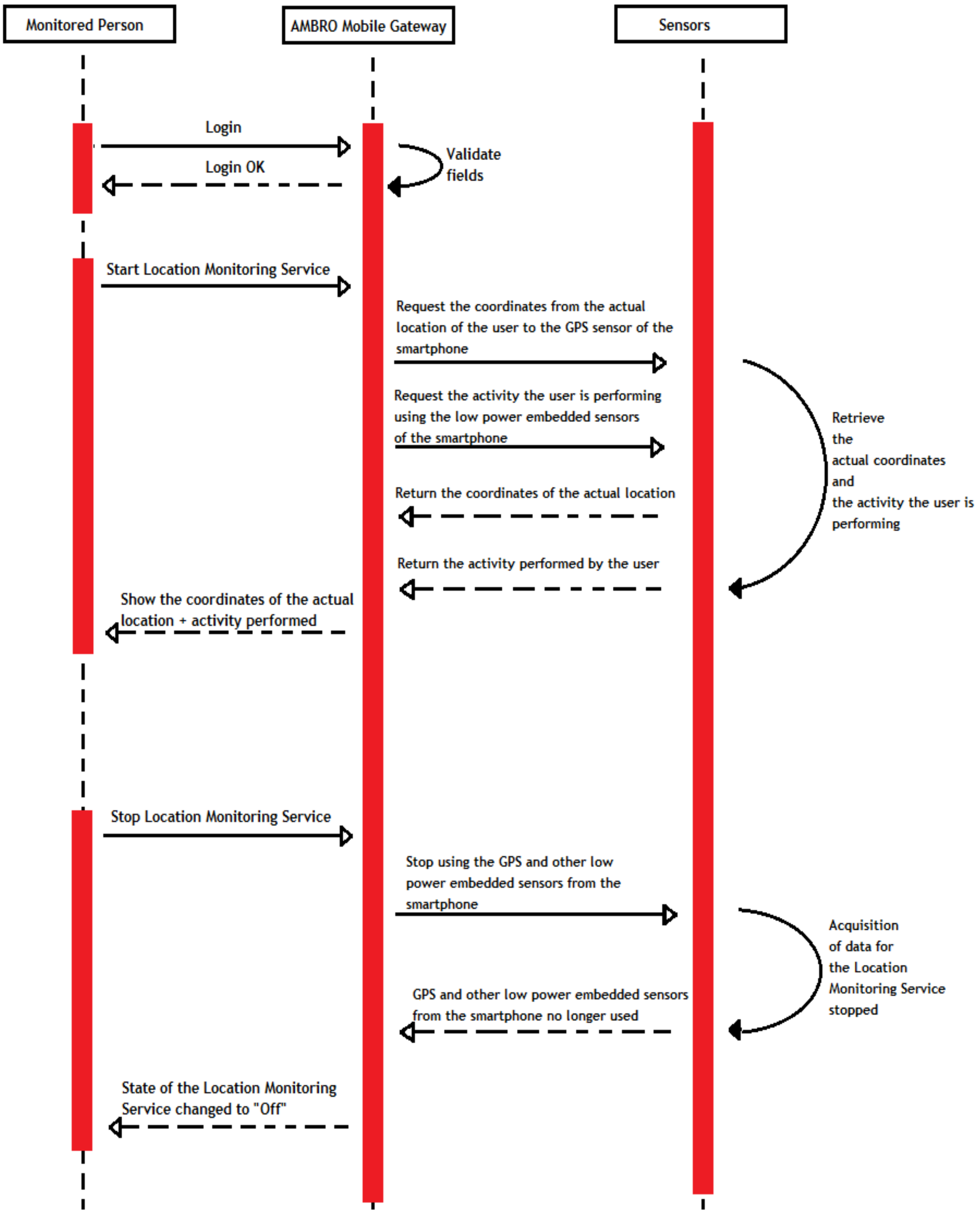

Fig. 24 - Location monitoring service sequence diagram.

The location monitoring service sequence diagram shows the actions performed on the AMBRO mobile gateway and respective sensors (GPS and other low power sensors embedded on the smartphone) used on the location monitoring service when the monitored user starts or stops the service.

Figure 25 illustrates the sequence diagram for the heart rate monitoring service.

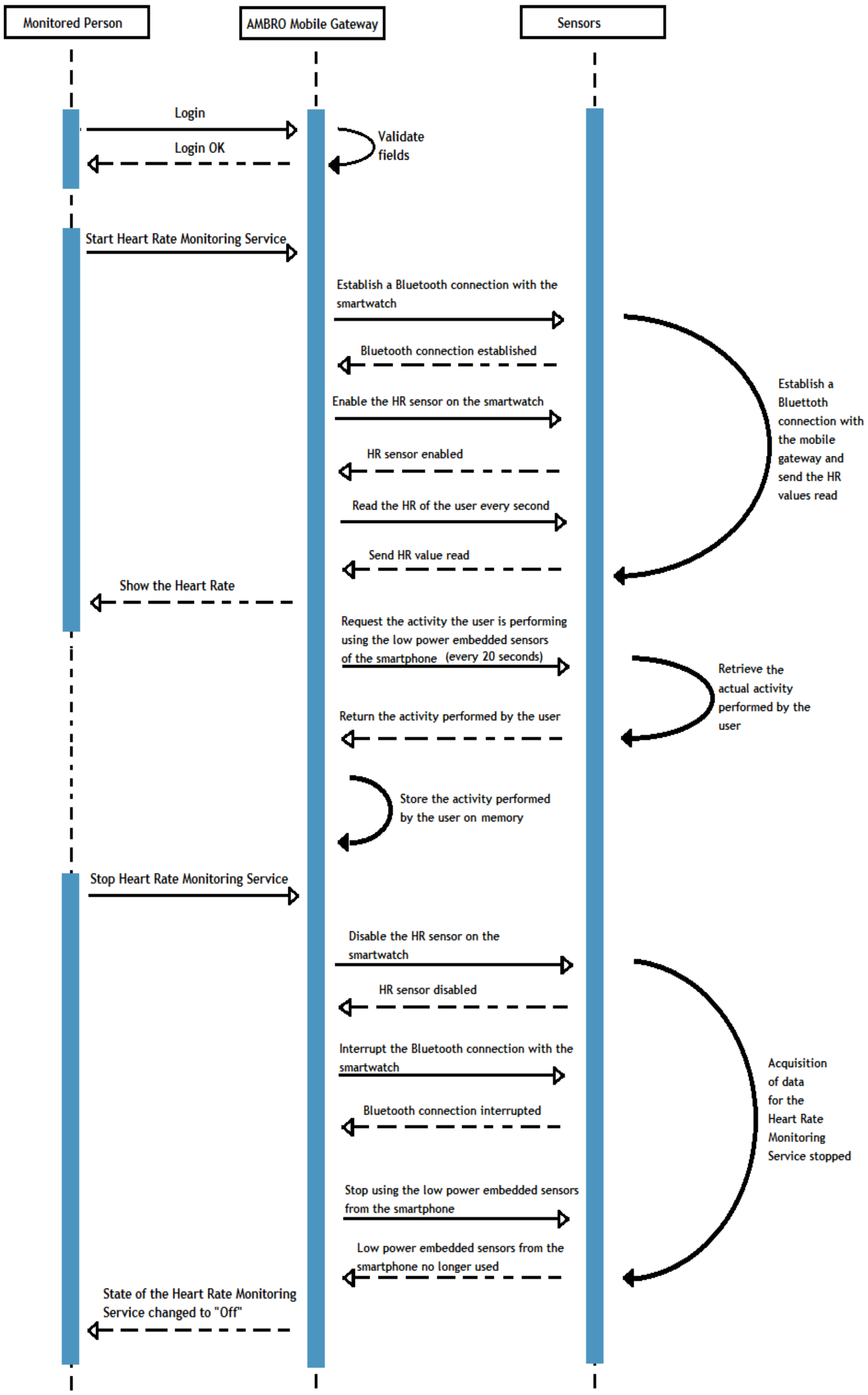

Fig. 25 - Heart rate monitoring service sequence diagram.

On the heart rate monitoring service sequence diagram are described the actions performed by the smartphone acting as mobile gateway, and the actions performed by the smartwatch when the user starts or stops the heart rate monitoring service.

Figure 26 shows the fall detection service sequence diagram.

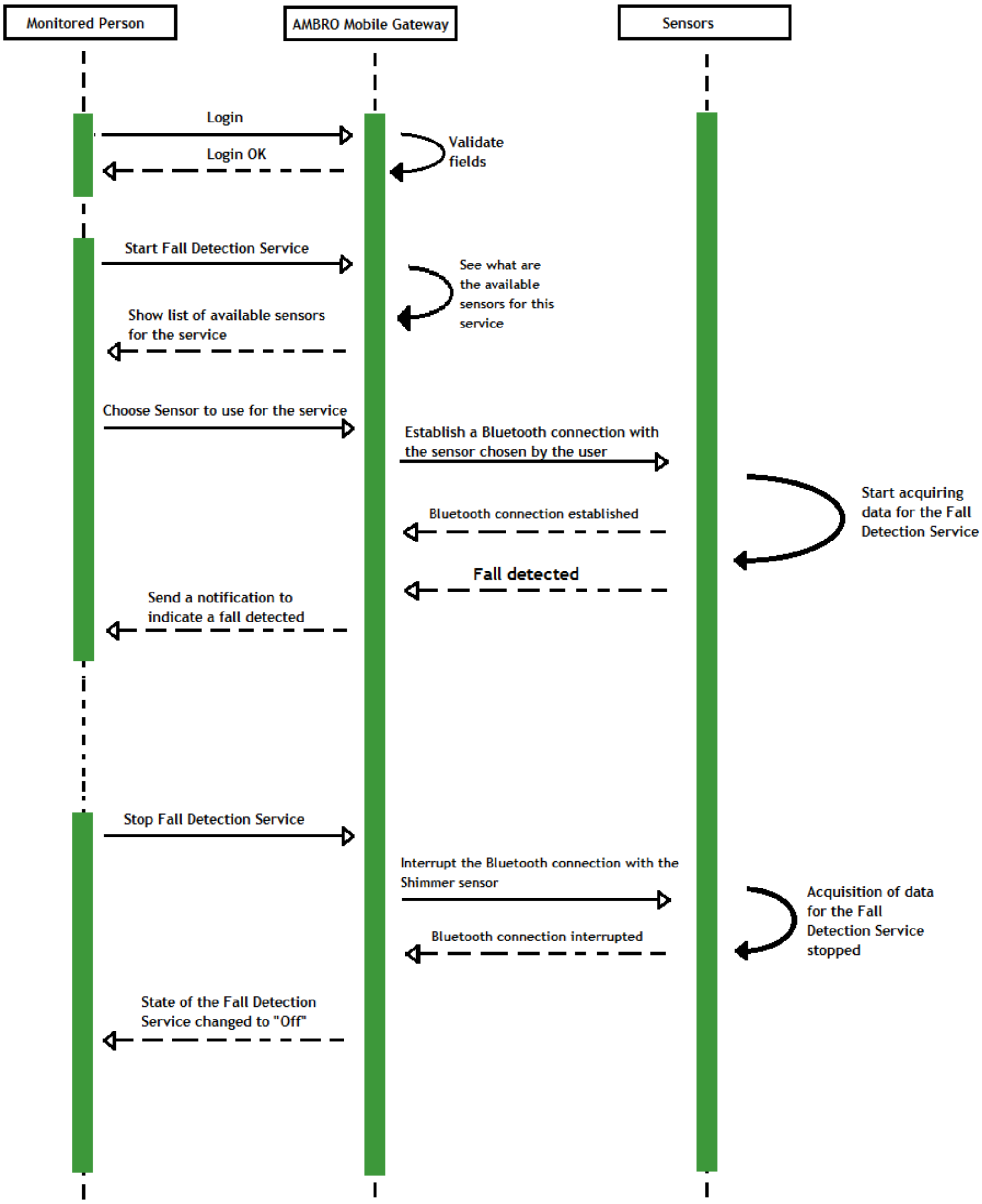

Fig. 26 - Fall detection service sequence diagram.

On the fall detection service sequence diagram are described the actions performed by the AMBRO mobile gateway and the actions performed by the Shimmer module when the monitored person starts or stops the fall detection service. There is also described what happens when a fall is detected.

Figure 27 shows how the mobile gateway notifies the IPA of the caretaker about relevant information related with the monitoring services.

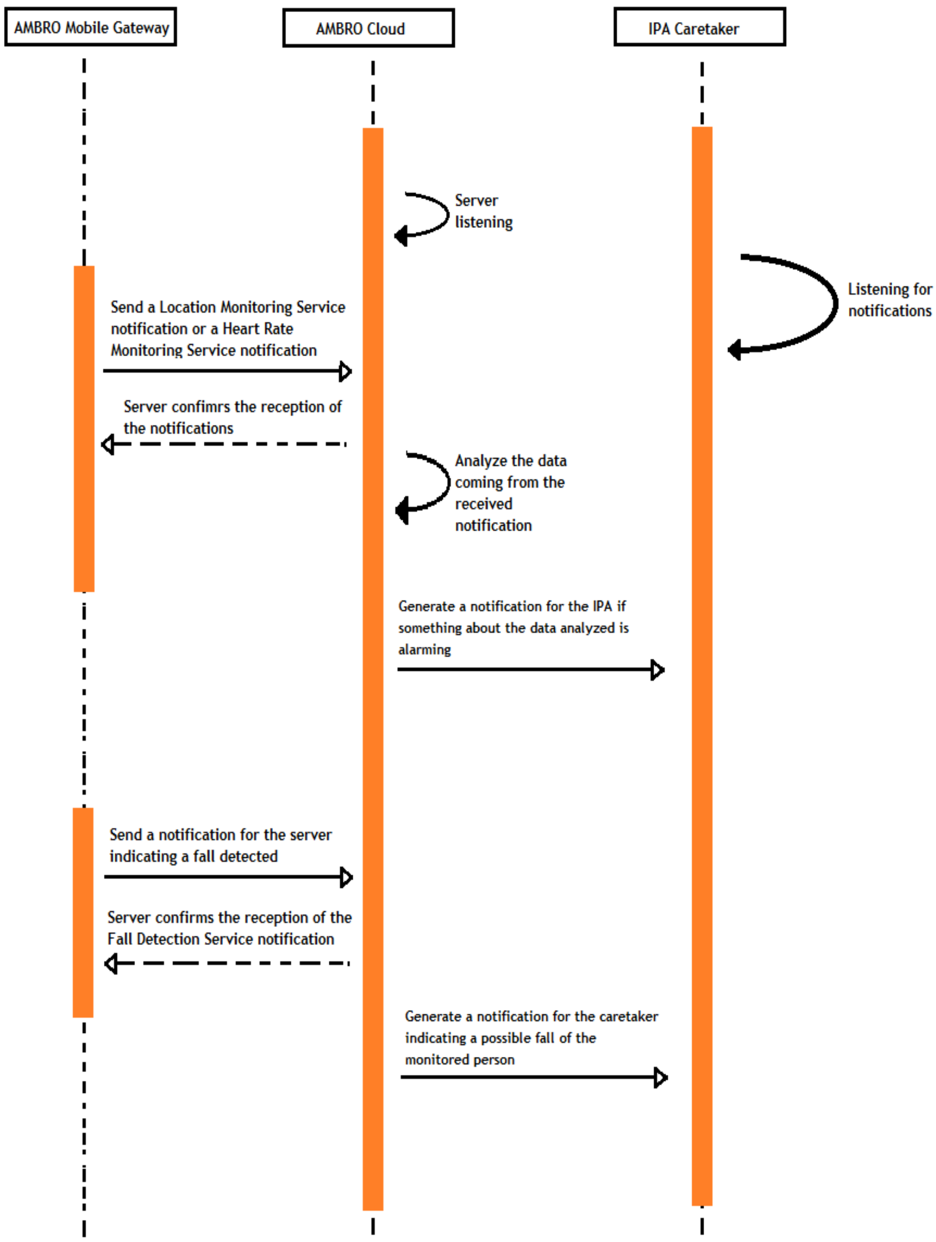

Fig. 27 - Sequence diagram when the mobile gateway as to notify the IPA about information associated with the monitoring services.

On the diagram presented on Figure 27 are shown the actions that happen in the system when the information from the mobile gateway reaches the AMBRO cloud. If the received information is alarming, the IPA of the caretaker is notified. Otherwise, no information is forwarded to the IPA.

4.3. Structural Diagrams

Structural diagrams give the information about the static structure of the system. It represents the classes, objects, interfaces, components and nodes of the system. On this chapter is presented the class diagram.

4.3.1. Class Diagrams

Class diagrams are the most common diagrams used in UML. Class diagrams are generally used on the development stage of the system, representing the object orientation of the system. Figure 28 shows the main class diagram of the mobile application module. It represents the classes, methods and objects used on the smartphone Android application that allow the monitoring services and activities to operate properly.

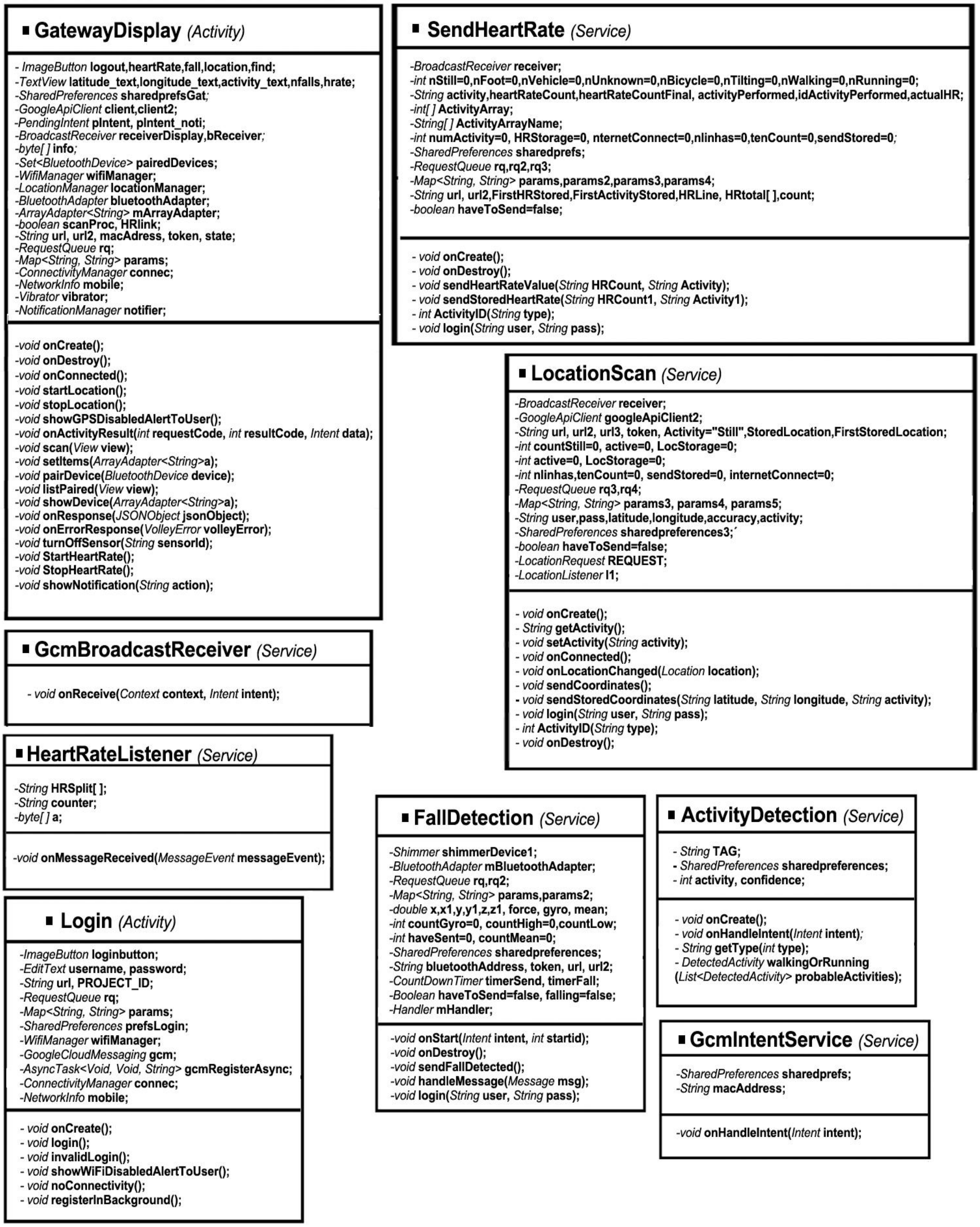

Fig. 28 – Main class diagram of the mobile application module.

The smartphone (mobile) module class diagram uses several classes. Some of those classes are used to show the graphical interface to the user. The class “Login” is the main class and only appears if the user has not yet logged in into the system. The other class is the “GatewayDisplay”, which controls all the buttons selectable by the user to enable or disable the monitoring services. The monitoring services are represented by different classes. The location monitoring service uses the “LocationScan” and the “ActivityDetection” classes. The fall detection service uses the “FallDetection” class. The heart rate monitoring service uses three classes: the “ActivityDetection” class to check the user activity, the “HeartRateListener” class to receive the information coming from the smartwatch, and the “SendHeartRate” class to send that information to the AMBRO cloud. The “GCMBroadcastReceiver” and the “GCMIntentService” classes are used to listen for push notifications coming from the server in order to start or stop the monitoring services. Those two classes are necessary to let the caretaker remotely control the services of the mobile gateway.

There are classes used on the smartphone (mobile) module responsible to assure the security of the communication. The class diagram of those classes is represented on Figure 29.

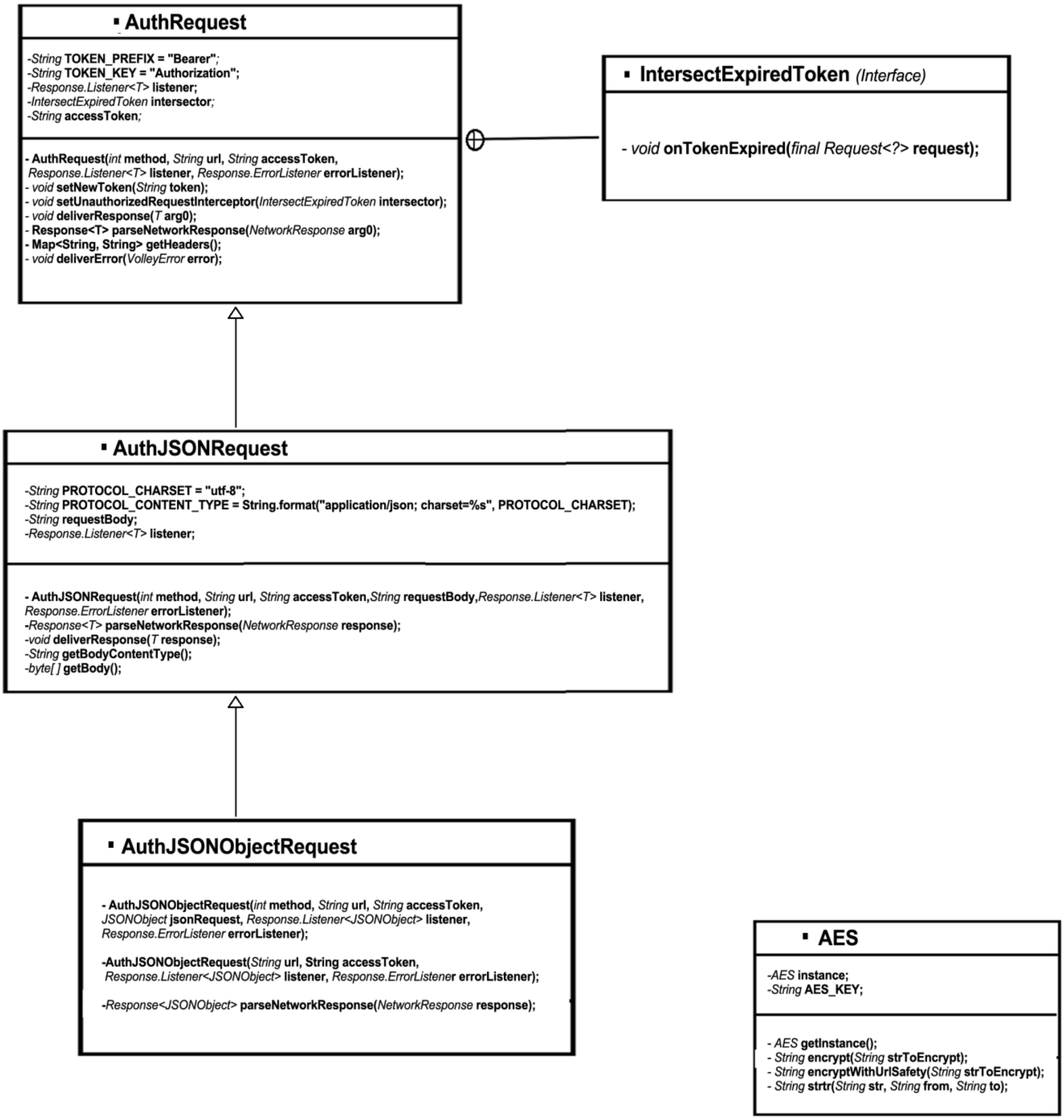

Fig. 29 – Class diagram of the security classes of the mobile application module.

The “AES” class is used to handle the data encryption, such as passwords. The other three classes are used to authenticate data sent to the AMBRO cloud through Web services. These classes are the

“AuthRequest”, “AuthJSONRequest” and “AuthJSONObjectRequest” classes.

There is also a class diagram for the wearable module of the application. It represents the classes used to develop the AMBRO mobile gateway application for the smartwatch. Figure 30 shows the wearable module class diagram.

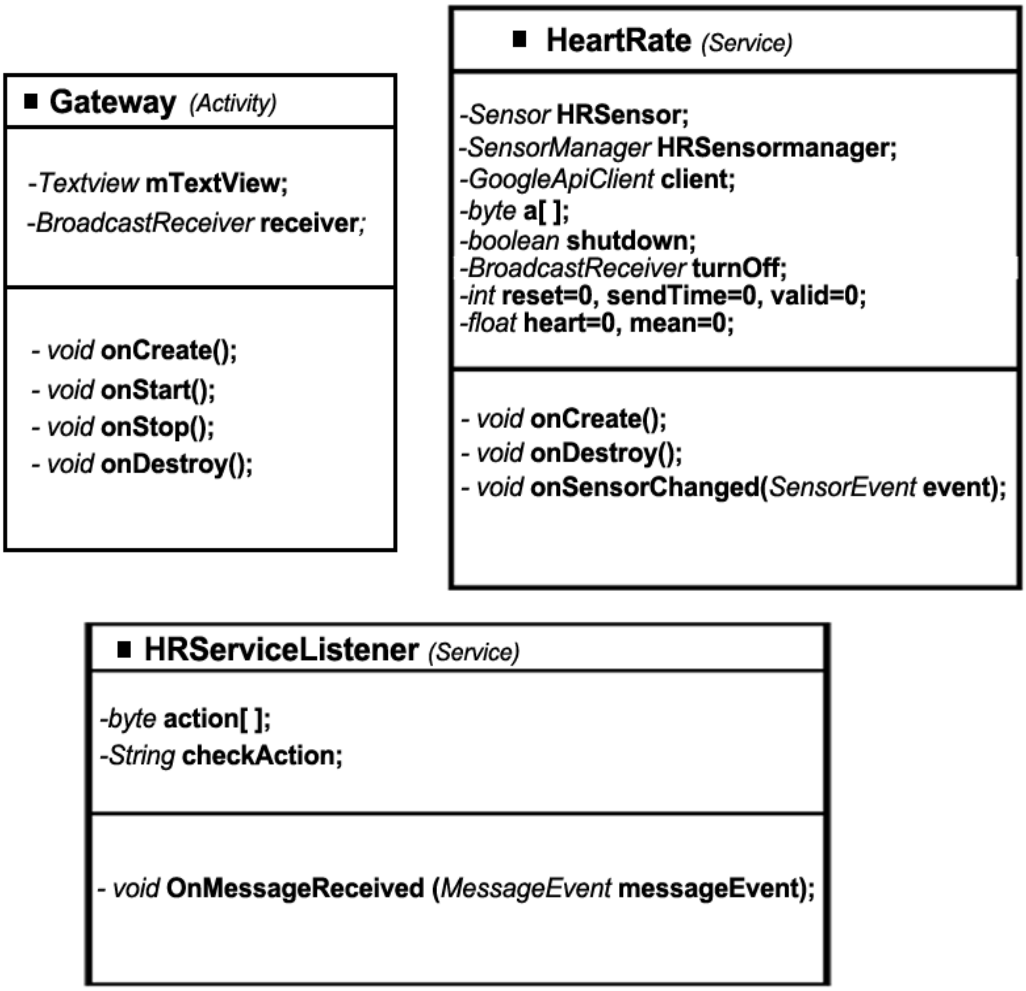

Fig. 30 - Class diagram of the wearable application module.

The class diagram of the wearable module of the AMBRO mobile gateway application has three classes. The “Gateway” class is used to exhibit the heart rate values on the smartwatch display. The “HeartRate” class has the purpose of listening to the information coming from the optical heart rate sensor, being also responsible to send that information to the AMBRO mobile gateway. The “HRServiceListener” is a class that extends a service that is always listening for possible messages coming from the mobile gateway. Those messages are used to enable or disable the heart rate sensor.

4.4. Summary

This chapter presented the requirement analysis for the AMBRO mobile gateway application. On Section 4.1 were presented the essential requirements of the system. Section 4.2 introduced the behavioural diagrams, particularly the activity, use case, and sequence diagrams. Section 4.3 presented the structural diagrams, with the class diagrams for the smartphone and smart watch modules of the application being described. Lastly, this section summarizes the chapter 4.

5. Devices and Technologies Used on the Prototype

The AMBRO mobile gateway application is used on a mobile health monitoring scenario including a set of sensors are responsible for monitoring a person in real time. This chapter presents the devices used for developing the AMBRO mobile gateway application. It is described the importance of each device for the system, including the technologies used by them.

5.1. Android Smartphone

This section will describe the Android smartphone used on this project. It will be explained its operation while performing the mobile gateway role, and while its operation as a tracking system. It is also described the Android OS and its main components.

5.1.1. Mobile Gateway

The number of persons owning a smartphone has increased in the last few years. Smartphones can be seen as the combination of several gadgets and electronic consumer devices on a single handheld device carried by people everywhere.

Within mobile communication scenarios, as the ones composed by mobile sensors, it can be necessary to use a mobile gateway to forward the traffic generated by mobile sensors to the Internet. It is the case of ubiquitous communication scenarios, where IoT communication environments can be referred. Therefore, smartphones appear as one of the best possibilities to perform the mobile gateway role, since they are devices used on a daily basis, affordable, powered by batteries, capable to communicate with other devices through Bluetooth or BLE (on some cases), and capable to access the Internet through Wi-Fi, GPRS, 3G or LTE networks (on most cases).

To this project, it was chosen the Samsung Galaxy Express 2, equipped with the Android OS 4.4.2, CPU dual-core 1.7 GHz, internal memory up to 8 GB , 1.5 GB RAM, BLE connectivity, and Wi-Fi, 2G, 3G, 4G and LTE wireless network connectivity. This smartphone fulfilled all the requirements identified as necessary to play the AMBRO mobile gateway role on the AMBRO project. Also, the Android OS version of the Samsung Galaxy Express 2 is higher than the minimum version required to pair an Android smartphone with an Android Wear smartwatch, allowing the communication between the two devices (necessary for the heart rate monitoring service).

5.1.2. Tracking System

Smartphones are nowadays equipped with a great number of sensors that can detect movements, environmental conditions, and physical positions. The combination of all the sensors embedded on smartphones and the different location providers available, such as GPS, network and cell tower providers, can enable the creation of intelligent tracking systems. Intelligent tracking systems are used to accurately give the current position of the smartphone, and therefore, the current position of its owner.

On the AMBRO mobile gateway application, the location monitoring service used to track the position of the person being monitored was developed using the GPS system and other low power embedded sensors of the mobile gateway smartphone. It was taken advantage of the user necessity to always carry the smartphone to make the system fully functional. By using its GPS system, the smartphone assumed a double role on the project: one part as a gateway of the system, serving only as a communication channel between the sensors used on the monitoring services and the AMBRO system; and other part as a smart object, that tracked simultaneously the position and the activity performed by the monitored person.

5.2. Android Smartwatch

The wearable technology is a booming trend nowadays. It is composed by a great diversity of devices, such as bracelets, caps, watches, shirts, and glasses, among other devices. Many of those wearable devices can measure physical condition indicators of their users, such as the heart rate or skin temperature. Those indicators can be important for fitness or healthcare applications. Because of their size and portability, wearable devices can be thought as full time monitoring devices.

The heart rate monitoring service available on the AMBRO mobile gateway application was designed taking into account the use of a device able to measure the heart rate of the user while being as less uncomfortable and intrusive as possible at the same time. Thus, a smartwatch was chosen, since watches are devices used by a lot of people on their quotidian. Besides, a smartwatch can measure the heart rate of the user while he is using it for other purposes. An Android Wear smartwatch was chosen because the Android Wear OS is open source. Also, the development of applications for the Android Wear Os is a process very similar to the process involved on the deployment of applications for the Android OS.

The smartwatch used on this project was the Moto 360, equipped with the Android Wear OS (version 5.0.2), a Texas Instruments OMAP 3 processor, 4 GB of internal memory and 512 MB RAM. It has BLE connectivity, and two embedded sensors: an optical heart rate sensor and a pedometer. Figure 31 shows a real image of a Moto 360 smartwatch.

Fig. 31 - Moto 360 smartwatch front (left) and Moto 360 smartwatch back with optical heart rate sensor enabled (right).

5.3. SHIMMER Sensing Platform

Shimmer is a wearable sensor platform developed by Intel in 2006, and used by researchers to develop prototype research applications. It includes wireless sensors for different scopes, such as kinematic (accelerometer, gyroscope), biophysical (electrocardiograph, galvanic skin response) and ambient sensing (temperature, light). It is an open source platform, flexible and highly configurable that allows the use of different radio communication technologies, as Bluetooth and the IEEE 802.15.4.

To develop the fall detection service available on the AMBRO mobile gateway application it was necessary to choose a sensor module capable of detecting falls. By using the Shimmer Sensing Platform it was possible to combine an accelerometer and a gyroscope on a unique sensor module. The attachment of the sensor module to the user made possible the measurement of the acceleration and the angular velocity of the user in real time, and checking if he/she was falling or not.

The Shimmer Sensor Platform allows the development of wireless and low power applications using a Texas Instruments MSP430 microcontroller unit. The built-in CC2420 radio transceiver allows Shimmer modules to communicate over 802.15.4 radio. Shimmer devices communicate over Bluetooth (class 2) with other devices. Thus, on the AMBRO mobile gateway application, the Shimmer module communicates with the mobile gateway over a Bluetooth connection.

Shimmer also allows the connection of external expansion daughter boards to connect the kinematic, biophysical and ambient sensors to the main board. It is also possible to store data on a microSD card up to 2GB. The firmware used on the Shimmer platform runs on TinyOS, which is a lightweight operating system, event oriented, generally used on WSN nodes. The development of Android applications with Shimmer devices is eased by the use of an Android API provided by the Shimmer itself.

The Shimmer kit used for this project was the Shimmer2R equipment, which included Shimmer modules with embedded 3-axis accelerometers and gyroscopes, and expansion boards to connect 3-lead electrocardiogram (ECG), electromyography (EMG) and galvanic skin response (GSR) sensors. The kit also included a Span USB 802.15.4 Radio Adapter to connect to USB ports on the PC and a USB-reader dock programmer/charger to program and charge Shimmer modules available on the kit. Figure 32 shows an example of a Shimmer device.

Fig. 32 - Example of two Shimmer modules: without any expansion board (left) and with a GSR expansion board connected (right).

5.4. Used Technologies

On the conception of the AMBRO mobile gateway application was used a smartphone, as it is a device able to communicate with other devices on high mobility scenarios. Then, it was necessary to aim at a specific mobile operating system. The Android operating system (OS) was chosen, as it is open source and widely adopted by the software development community. However, other mobile operating systems such as the iPhone Operating System (iOS) and the Windows Phone could be adopted through the use of the necessary programming languages for those both cases. Although, the same methodologies and mechanisms used on the AMBRO mobile gateway application for the Android OS could be used with other mobile operating systems.

The Java programming language was used to develop the AMBRO mobile gateway application for the Android OS. The Android Studio IDE [101] was the tool chosen to build the application, providing a software development kit with pre configured libraries, emulators and debugger tools. The Android Studio IDE can be used to develop a single project with different types of modules, allowing the automatic synchronization between smartphone and wearable devices applications belonging to the same application package. It was helpful for this project as it allowed the development of the main mobile gateway application for an Android smartphone, and a secondary application for an Android Wear smart watch.

5.4.1. Android OS

The Android OS is used in mobile devices based on Linux and developed by the Open Handset Alliance, led by Google among other enterprises. Although been founded in 2003, the first Android OS device commercially available only appeared in 2008, with the announcement of the G1 HTC Dream, a T-Mobile’s smartphone. Billions of devices equipped with the Android OS have been released and sold in the market until the year of 2015, making it the most used mobile operating system. Despite being mostly used on smartphones, the Android OS can also be used on other electronic devices, as computer tablets, digital cameras, home appliances, glasses, wristwatches, and even cars and TVs with specialized interfaces.

The Android OS architecture is organized on a software stack composed by five different sections and four main layers. Each layer of the stack is completely integrated with other layers, and providing an optimal environment for the development of Android applications. Figure 33 illustrates the Android OS architecture.

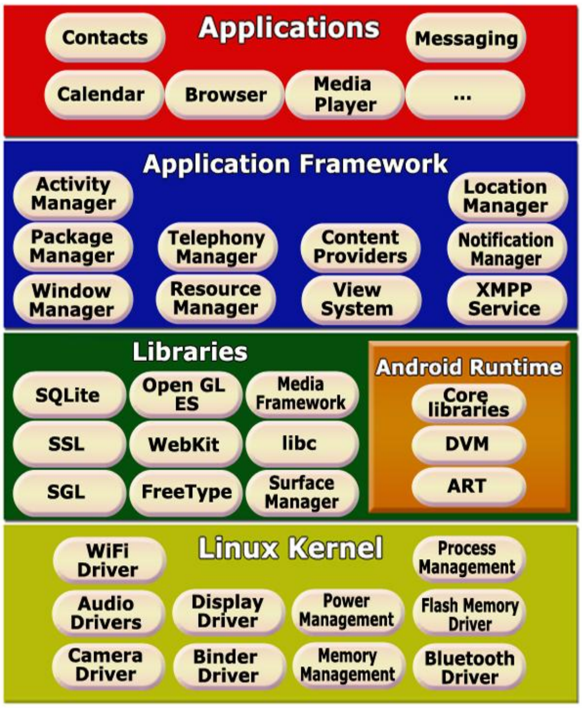

Fig. 33 - Android OS architecture.

The bottom layer of the architecture is the “Linux Kernel” layer. It is based on Linux and provides a level of abstraction between hardware devices and other layers of the Android architecture positioned at higher positions. The “Libraries” layer is composed by all the libraries necessary to develop Android applications, such as the libraries to build user interfaces, draw graphics and access databases. There is another section on the “Libraries” layer, denominated “Android Runtime”, which is comprised by three modules: (i) the Java core libraries, which support tasks such as string handling, file manipulation or networking, (ii) the Dalvik Virtual Machine (DVM), used as a Java Virtual machine specialized for Android applications, and the (iii) Android Runtime (ART), also used as a virtual machine for Android environments, which began to be used on Android devices since the version 4.4 of the operating system. The “Application Framework” layer provides a set of services that allow Android applications to be run and managed. The “Applications” layer is the higher layer of the software stack of the Android OS, comprising the core Android applications and other third party applications, allowing the installation of new applications on Android devices.

Android applications are composed by one or more type of components. Those components can be activated individually or through other applications. There are four main components on an Android application: (i) activities, (ii) services, (iii) broadcast receivers, and (iv) content providers. Only content providers are not used on the AMBRO mobile gateway application.

Activities represent the display of an application with a graphical user interface. An Android application can have none or multiple activities, depending on the number of different activity screens supposed to exist. Despite the option to have more than one activity, an Android application can only present a single activity at a single moment. The activity lifecycle management is executed through the implementation of callback methods. An activity is active if it is running on the foreground. An activity is paused when it is visible but loses focus to another process. When an activity is no longer visible, it is stopped; and when it is released from the system memory it is destroyed. On the AMBRO mobile gateway application there are two different activities. The first activity is responsible for the login process. If the login is successful, the main activity of the application is shown up, and the login activity is destroyed. The main activity allows the user to start all the services available on the AMBRO mobile gateway application. If the user logs out of the application, the main activity is killed and the login activity is started again.

Services are the Android components usually used when it is necessary to execute processes on the background. Typically, services do not interact directly with a user interface. There are used services running in the background to monitor the user of the AMBRO mobile gateway application. Those services are able to detect the activity that the user performs, his location, his heart rate and potential falls. All those services can be started on the activity responsible for displaying the main screen. Depending on the indicators that the user wants to monitor, the services can all run simultaneously, or one at a time.

Broadcast receivers are Android components that listen and receive system application notifications. Every receptor registered to a certain event is notified as soon as the event happens. On the AMBRO mobile gateway application there are broadcast receivers listening to some system events, as the ones used to change the aspect of the user interface every time a service is turned on or turned off, and others used to listen to messages coming from the AMBRO back office with the intuition of enable or disable the monitoring services remotely.

5.4.2. Android Wear OS

In 2014 Google announced a version of the Android OS for wearable devices, such as smart watches. The Android Wear is based on the Google Now service, being defined as an extension of the Google Search tool, by using a natural language processing interface to respond to questions made by the user, and by making recommendations using various Web services, as weather and location services.

Smart watches running the Android Wear operating system are essentially used to show notifications generated on a smartphone. Firstly, the smartphone (that has to run at least the version 4.3 of the Android OS) is paired with the smart watch via a BLE connection. Then, notifications generated on the smartphone can be shown on the smart watch. Those notifications can be accepted or rejected as the user finds appropriate. One example is when there are incoming calls. It has to be referred that all the possible actions that can be taken are finished on the smartphone.

On the AMBRO mobile gateway application, the use of a heart rate sensor embedded on a smart watch running the Android Wear OS allowed the detection of the heart rate of the user in real time. The information regarding the heart rate of the user is sent via BLE to the mobile gateway in order to be forwarded to the AMBRO cloud. It was necessary to use the Wearable Message API [102] to send data from the smart watch to the smartphone. On the smartphone is implemented a wearable listener service to receive and extract useful information of the data sent by the smart watch. Figure 34 shows a diagram that explains how the smartwatch sends data to the smartphone.

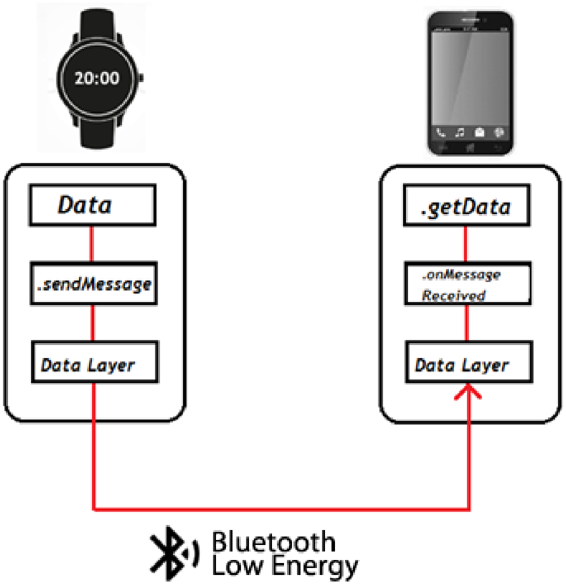

Fig. 34 - Illustration of the communication between an Android Wear smart watch and an Android smartphone.

One of the main advantages that the Android Wear OS brought to the developers was the possibility to develop open source applications for wearable devices in a similar way to the development of Android smartphone applications. The synchronization between applications developed for both smart watches and smartphones can enable the creation of larger and heterogeneous communication scenarios.

5.5. Summary

Chapter 5 focused on the devices and technologies used on the development of the AMBRO mobile gateway system. Section 5.1 described the smartphone used on the project. It is both used as a communication channel when used as a mobile gateway, and as a smart object when used as a tracking system. Section 5.2 presented the Moto 360 smart watch. On Section 5.3 the Shimmer sensor module, used to detect possible falls of its user is described. Section 5.4 enumerated the technologies used by the devices comprising the AMBRO mobile gateway application, including a description of the Android OS and the Android Wear OS. Finally, this section summarizes the chapter.

6.AMBRO Mobile Gateway

The AMBRO mobile gateway is an Android application aimed to mobile health scenarios where a person needs to be always under monitoring. The AMBRO mobile has three monitoring services available: the location monitoring service, heart rate monitoring service, and fall detection service. Those three services can be directly started by the AMBRO mobile gateway user, or remotely started by the caretaker on his AMBRO back office. A fourth service is available at the AMBRO mobile gateway and it is used to discover objects on the surrounding environment.

This chapter describes the AMBRO mobile gateway application, showing its importance to the AMBRO main system, and explaining each one of its functionalities.

6.1. AMBRO Mobile Gateway Architecture

This section presents the AMBRO mobile gateway architecture. On ubiquitous communication environments, devices can change their position constantly. Problems may arise on such scenarios if suitable solutions to allow the communication between mobile devices within the network are not developed. The mobile gateway on the AMBRO system is used on a mobile health scenario, allowing an AMBRO user to remotely monitor a person under his care. Figure 35 illustrates the AMBRO mobile gateway architecture, which includes four modules: the BSN, the Mobile Gateway, the AMBRO cloud, and the IPA of the caretaker.

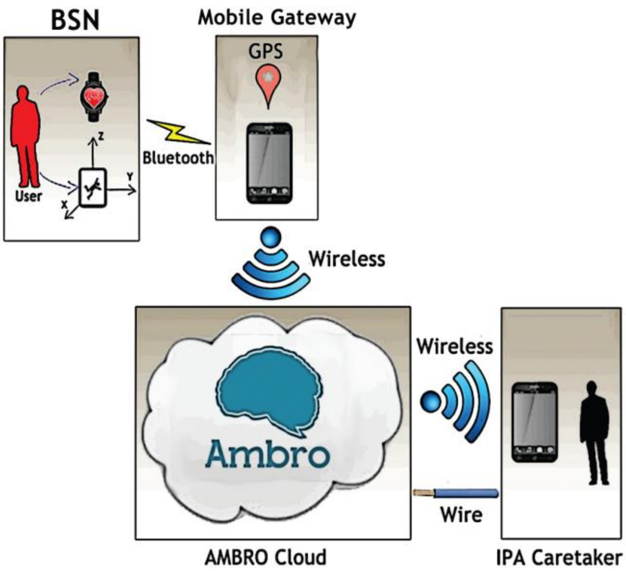

Fig. 35 - AMBRO mobile gateway system architecture.

The BSN module includes the sensors that monitor the user in real time. There are used two devices: (i) one Shimmer module using an accelerometer and a gyroscope to detect if the user fell, and (ii) a smartwatch with an embedded heart rate sensor to count the user heartbeat. Information gathered by both sensors is forwarded to the mobile gateway through a Bluetooth connection (the smartwatch uses BLE). The mobile gateway module is composed by a smartphone, which must be always carried by the monitored person. The smartphone must have the AMBRO mobile gateway installed to allow the user to control the monitoring services. The location monitoring service uses the smartphone GPS to determine in real time, and accurately the position of the monitored person. The information provided by the monitoring services is forwarded to the AMBRO cloud through RESTful Web services using HTTP. The information inside the AMBRO cloud is stored and processed by the multiple modules there existent and described on Chapter 3. A notification is generated to the IPA of the caretaker through a *push* notification mechanism if some indicator about the monitored person is alarming.

6.2. Graphical User Interface

This section presents the AMBRO mobile gateway application graphical user interface and its main features. The user needs an AMBRO account to use the application. On this particular case, the user can have an IPA and a mobile gateway associated to his account. The caretaker has the IPA on his possession, while the mobile gateway belongs to the monitored person. Figure 36 shows the splash screen of the AMBRO mobile gateway application, on which the user has to introduce his AMBRO account credentials.

Fig. 36 - AMBRO mobile gateway login screen.

The user introduces its credentials on the associated fields, and hits the “Login” button. If the login is successful, the main screen is shown. Figure 37 illustrates the AMBRO mobile gateway main screen.

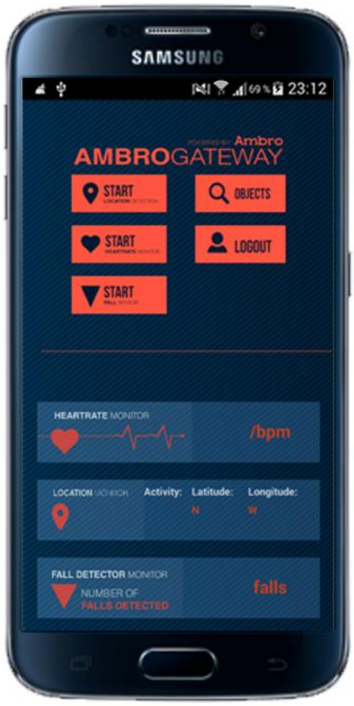

Fig. 37 - AMBRO mobile gateway main screen.

At the main screen the user can start the monitoring service, the discovery of Bluetooth objects on the environment, and logout from the application. Information regarding the monitoring services, such as the heart beats per minute, GPS coordinates and activity detected, and number of falls detected can be visualized on the bottom of the application main screen. While the AMBRO mobile gateway application is running, the user can press the home button or the back button of its smartphone without changing the actual state of the monitoring services. So, if a monitoring service is running when the user presses the back button, the service will keep running on the background. The application splash screen will be displayed again if the user presses the “Logout” button. By implementing this mechanism, all services that might be running on the application will be stopped.

The login operation on the AMBRO mobile gateway application is transparent to the user. A token is generated when the login is successfully made, allowing the communication between the mobile gateway and the

AMBRO cloud. The token is refreshed periodically until the user logs out from the AMBRO mobile gateway application. This mechanism assures the continuous monitoring of the person, with data being sent to the cloud even on scenarios where the application screen is not shown on the smartphone display.

6.3. Location Monitoring Service

The location monitoring service uses the smartphone GPS to retrieve the user position. It also discovers the activity that the user is performing by using the low-power embedded sensors on the smartphone. The Google's Activity Recognition API [103] was used to discover the activity, detecting if the user was still walking, running, driving a vehicle or riding a bike. Figure 38 shows the application display when the location monitoring service is running.

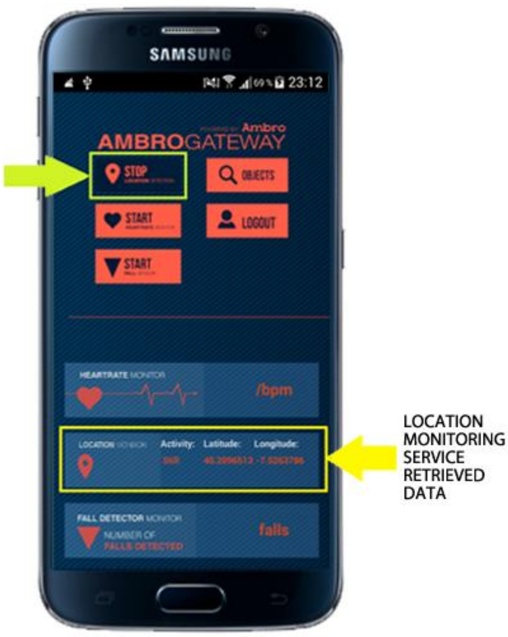

Fig. 38 - Application display with the location monitoring service running.

It is necessary to have the smartphone GPS enabled when the application is started. Also, the system automatically enables the Wi-Fi on the smartphone if the user has not enabled it previously. On scenarios where the GPS was not enabled, a dialog box appears to the user asking him to go to the smartphone options menu to enable the GPS. If the user answers positively to that question, he is redirected to the options menu, where he can enable the GPS system on the smartphone. When the GPS is enabled, the color and text of the location monitoring service button are changed to indicate that the service is running. The coordinates and activity detected by the low power embedded sensors of the smartphone are shown on the bottom of the application display.

6.4. Heart Rate Monitoring Service

The heart rate monitoring service is used to check the heart rate of the AMBRO mobile gateway user. When the service is started, the smartphone tries to establish a connection with the smartwatch through BLE. Figure 39 shows the application display when the heart rate monitoring service is running.

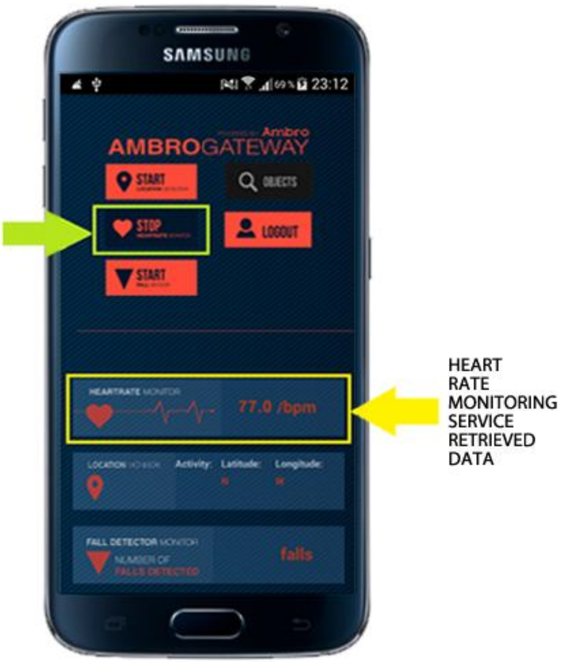

Fig. 39 - Application display when the heart rate monitoring service is running.

When the service is started, the Wi-Fi and the Bluetooth need to be enabled to make the service run. If they were not enabled, the system automatically enables them. When the service starts, the color and text on the heart rate monitoring service button are changed, and the user heart rate is shown on the bottom of the application display. Also, the “Objects” button is disabled when the heart rate monitoring service is running, preventing the smartphone to send Bluetooth broadcast messages to the environment when it already has a Bluetooth connection established with the smartwatch. This mechanism avoids the deterioration of the communication between the smartphone and the smartwatch.

6.5. Fall Detection Service

The fall detection service is used to detect if the user fell. When the user clicks on the “Start Fall Sensor” button, he has to choose a Shimmer device to establish the connection with the smartphone. Both Wi-Fi and Bluetooth have to be enabled on the smartphone. If they were not previously enabled by the user, the system automatically enables them. Figure 40 shows the application display when the user tries to start the service.

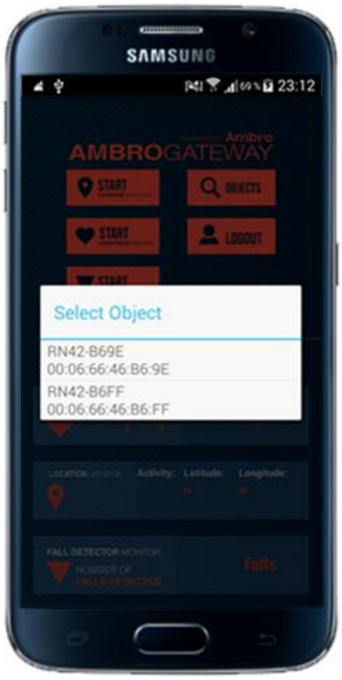

Fig. 40 - Application display when the user tries to start the fall detection service.

The smartphone tries to establish a Bluetooth connection with the Shimmer device chosen by the user. If the connection is successful, the Shimmer device starts the streaming, and the fall detection service button changes its appearance, presenting another color and text. When a fall is detected, the information is displayed at the bottom of the application display. When the service starts, the “Objects” button changes its color to black and becomes disabled, preventing the smartphone to send Bluetooth broadcast messages to the environment when it already has a Bluetooth connection established with the Shimmer device. This mechanism is used to avoid the deterioration of the communication between the smartphone and the Shimmer device. Figure 41 shows the application display when the Fall Detection service is running.

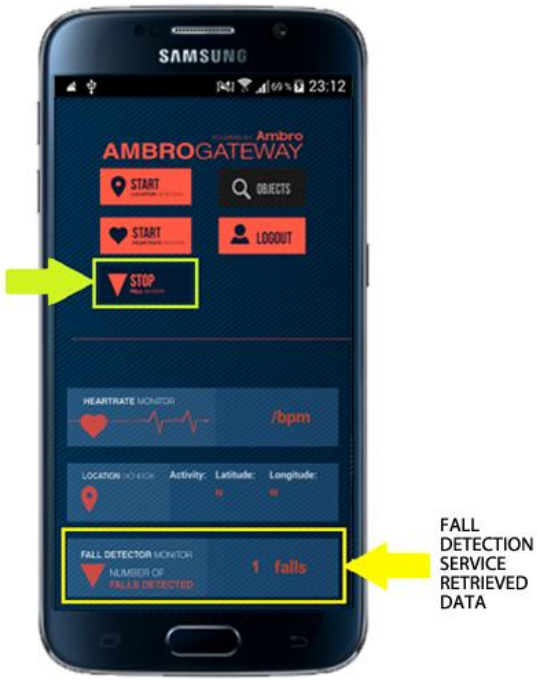

Fig. 41 - Application display when the fall detection service is running.

6.6. Object Search Service

The object search service is used to discover sensors that communicate over Bluetooth, such as the Shimmer devices. The Bluetooth has to be enabled on the smartphone when the service is started. Otherwise, the system itself enables the Bluetooth on the smartphone automatically.

To discover Bluetooth devices, the mobile gateway has to initially send Bluetooth broadcast messages to the environment. When the discovery process is completed, a list of Bluetooth devices that answered positively to the broadcast message is returned to the user. Later on, the user can select one of the discovered devices to pair with the gateway. If the device that has been paired with the gateway is a Shimmer sensor, it will be eligible to be used on the fall detection service.

The object search service can only be launched if the heart rate monitoring service and the fall detection service are not being executed.

This mechanism prevents the degradation of the communication between the mobile gateway and other Bluetooth devices, since it does not allow the communication environment to be burdened by Bluetooth broadcast messages. Figure 42 shows the AMBRO mobile gateway display after the object search service has been launched.

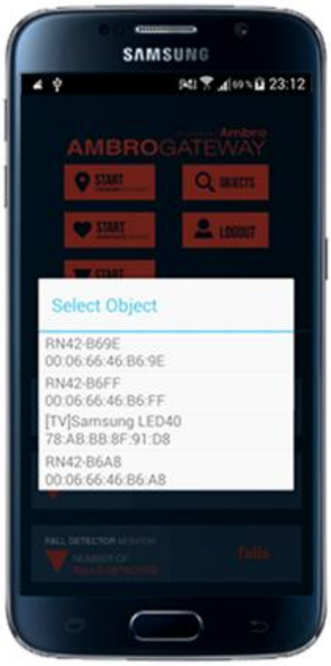

Fig. 42 - Application display when the Object Search service is running.

6.7. Monitoring Services Remote Control

The three monitoring services available on the AMBRO mobile gateway application can be remotely enabled and disabled by another user. It is useful because it allows the caretaker to have the total control of the AMBRO mobile gateway. Therefore, the caretaker can choose how he wants to monitor the person under his care, by enabling and disabling the AMBRO mobile gateway services as he deems necessary.

There are two things necessary to remotely control the AMBRO mobile gateway monitoring services: (i) the AMBRO mobile gateway must be connected to the Internet, and (ii) the caretaker has to access the AMBRO back office management tool. Then, the caretaker accesses the AMBRO back office and chooses the service (or services) to enable or disable. This information flows to the server existent on the AMBRO cloud, which then, informs the AMBRO mobile gateway about the services the caretaker wants to enable or disable.

The remote control of the AMBRO mobile gateway monitoring services uses the Google Cloud Messaging for Android (GCM) [104]. The GCM is a free service that allows servers to send data to GCM servers. Then, the GCM servers spread the messages by Android client applications there registered using a *push* mechanism. The client applications (the AMBRO mobile gateway on this case) do not need to run on the system foreground to receive notifications. To receive notifications, the client applications need to be registered on the GCM server and have an active listener to catch push notifications aimed to them. Figure 43 illustrates the GCM mechanism used in the AMBRO mobile gateway application.

Fig. 43 - Illustration of the push notifications mechanism on the AMBRO system.

A notification (accompanied with a vibration) is displayed on the smartphone screen every time the AMBRO mobile gateway receives a request to enable or disable one of the monitoring services. Each service generates a different notification. In Figure 44 is depicted an example where the AMBRO mobile gateway shows two notifications: the first notification indicates that the location monitoring service has been remotely enabled, while the second notification indicates the disabling of the location monitoring service by the caretaker.

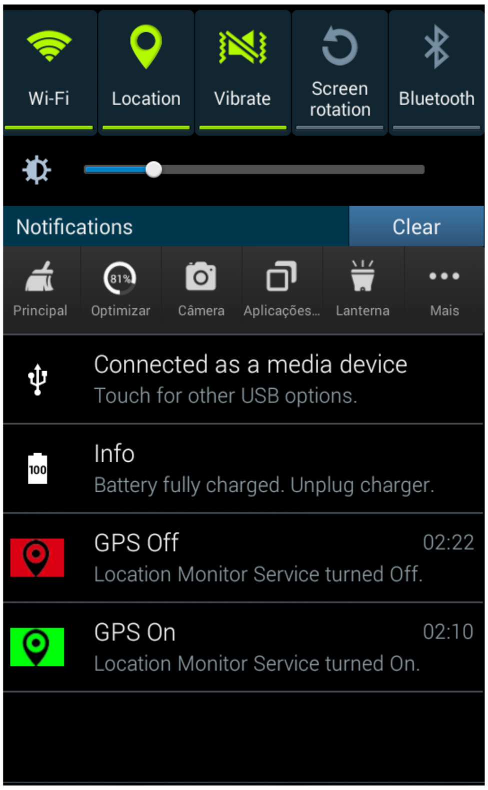

Fig. 44 - Example of a location monitoring service push notifications.

If the GPS was already enabled on the smartphone, the location monitoring service starts to gather the information about the location of the AMBRO mobile gateway user. If the GPS was not enabled, the AMBRO mobile gateway user has to enable it manually.

In Figure 45 is depicted an example where the AMBRO mobile gateway displays two notifications informing the user about the enabling and consequent disabling of the heart rate monitoring service by the caretaker is depicted.

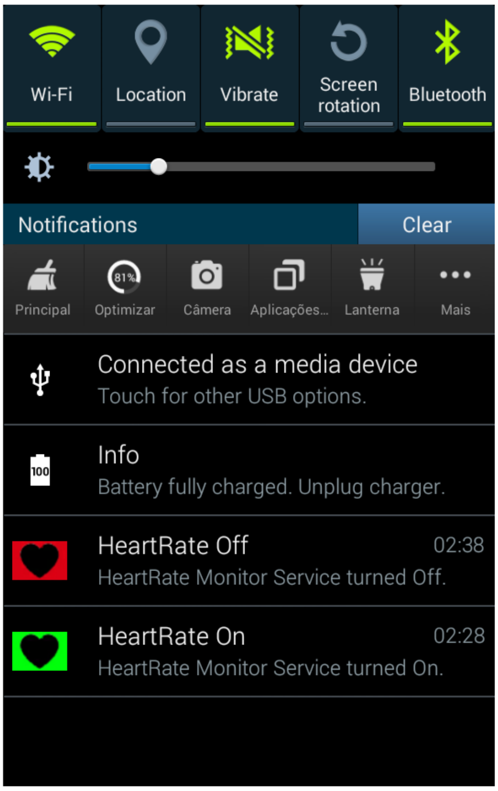

Fig. 45 - Example of an heart rate monitoring service push notifications.

On the heart rate monitoring service scenario, the Bluetooth needs to be enabled to make the service work. Unlike the GPS system, on Android devices, the Bluetooth adapter can be enabled automatically, being this situation transparent to the user.

Figure 46 illustrates an example where two notifications concerning the fall detection service are shown on the AMBRO mobile gateway application. On the first notification the user is informed that the fall detection service was remotely enabled by the caretaker. The second notification informs the user that the fall detection service was remotely disabled by the caretaker

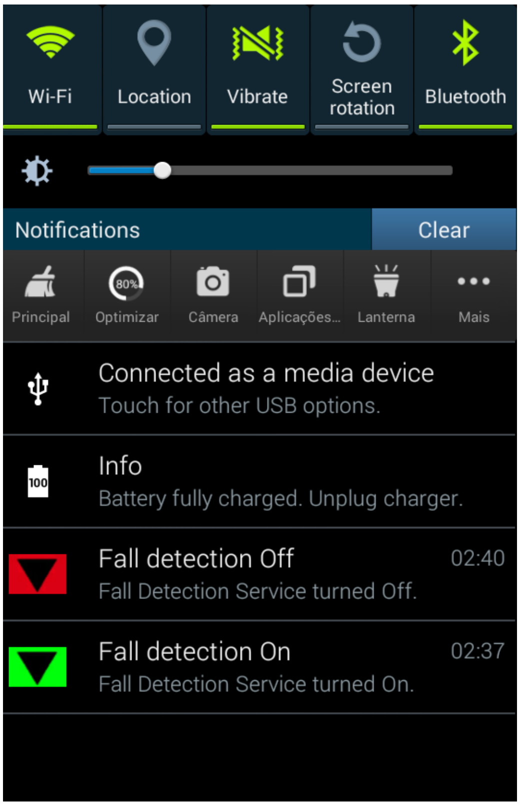

Fig. 46 - Example of fall detection service push notifications.

Just as the heart rate monitoring service, the fall detection service also needs the smartphone Bluetooth adapter enabled to correctly operate.

It has to be emphasized that the push notification messages received on the smartphone have a field specifying the MAC address of the device they want to access. By implementing such mechanism, the mobile gateway can identify the object that the caretaker wants to enable or disable.

6.8. Summary

On this chapter, all the AMBRO mobile gateway application features were presented. Section 6.1 presented the AMBRO mobile gateway architecture, describing the four modules that constitute the system. The main focus on that section was on the BSN and mobile gateway modules, which are directly inserted on this project scope. On Section 6.2 was presented the graphical user interface of the application, describing some features made available on each different screen display. Section 6.3 described the location monitoring service, showing its graphical user interface and its main features. Section 6.4 described the operation of the heart rate monitoring system, including the graphical user interface of the application when the service is running. Section 6.5 explained the fall detection service. On Section 6.6 was presented the object search service, which is used to detect Bluetooth objects eligible for the fall detection service. In Section 6.7 was explained how the caretaker can enable or disable the AMBRO mobile gateway monitoring services by using push notifications on the GCM mechanism. On that section, there were also shown some screenshots of the application when the push notifications were received by the mobile gateway. Finally, this section summarizes the chapter 6.

7. Performance Evaluation, Validation and Result Analysis

This chapter addresses the performance evaluation of the AMBRO mobile gateway. The performed experiments were based on three metrics: (*i*) power consumption on the system devices, (*ii*) accuracy of the results returned by the mobile gateway services, and (*iii*) communication with other objects. The power consumption was measured on the smartphone and on the smartwatch by varying some parameters regarding the monitoring services. By evaluating the accuracy of each monitoring service was possible to check if the services were operating as expected. The communication between the mobile gateway and other objects is supposed to demonstrate the construction of an ubiquitous communication scenarios between heterogeneous objects.

7.1. Power Consumption and System Performance

The power consumed by each device in IoT environments is very important to evaluate the system performance on such scenarios. This is due to the characteristics presented by IoT devices, such as high mobility, limited power supply, and capacity to communicate anytime and anywhere. Some strategies need to be applied to comply with all those conditions without degrading the system performance. For example, the batteries of the objects can last longer if they only communicate when it is strictly necessary. Different experiments were conducted on the AMBRO mobile gateway application to evaluate the parameters that were changeable to make the power consumption of the system optimized. For both the location monitoring and heart rate monitoring services different experiments were performed by simply varying the rate on which activity detection updates were requested. Another parameter which varied for each experiment performed on the heart rate monitoring service was the periodicity at which the data was sent to the AMBRO cloud. On the fall detection service, the reading frequency for the acceleration and for the angular velocity on the Shimmer module varied for each scenario where the AMBRO mobile gateway was tested. Table III summarizes the first experiment scenario for the AMBRO mobile gateway application.

Table III - AMBRO mobile gateway first experiment scenario.

Services	Tested Features					
	Activity Detection	*Check Heart Rate*	*Reading Frequency*	*Send Data to the Cloud*	*Smartwatch Battery*	*Smartphone Battery*
Location Monitoring Service	20 seconds	-------	-----	When a location update is requested	11h 15minutes	19h (approx.)
Heart Rate Monitoring Service	20 seconds	1 second	-----	5 minutes		
Fall Detection Service	-------	-------	20 milliseconds	When a fall is detected		

On the first test scenario, the activity of the user on the location monitoring and heart rate monitoring services was checked with a 20 seconds periodicity. The heart rate of the monitored person was measured every second, with that information being sent to the AMBRO cloud every 5 minutes. The Shimmer module operated on a 20 milliseconds rate to detect if the user has fallen. On the location monitoring service, the communication with the AMBRO cloud was not performed periodically, since location updates are not requested if the user is not in movement. The fall detection service only generated notifications when a fall was detected. Therefore, the communication with the cloud was only performed on those scenarios. On this experiment scenario, the smartwatch battery lasted 11 hours and 15 minutes and the smartphone battery lasted 19 hours approximately.

The second experiment scenario is summarized on Table IV.

Table IV - AMBRO mobile gateway second experiment scenario.

Services	Tested Features					
	Activity Detection	*Check Heart Rate*	*Reading Frequency*	*Send Data to the Cloud*	*Smartwatch Battery*	*Smartphone Battery*
Location Monitoring Service	40 seconds	-------	-----	When a location update is requested	11h 15minutes	19h 15 minutes (approx.)
Heart Rate Monitoring Service	40 seconds	1 second	-----	5 minutes		
Fall Detection Service	-------	-------	50 milliseconds	When a fall is detected		

There are some parameters that varied from the first experiment scenario to the second experiment scenario. Essentially, the activity detection rate on the second experiment scenario was twice when compared with the first experiment scenario. The variation of that parameter allowed the smartphone battery to be a little spared. On the fall detection service, the Shimmer module read the acceleration and the angular velocity at a 50 milliseconds rate. The heart rate of the user was checked every second, like in the first experiment scenario. Thus, the smartwatch battery lifetime on both experiment scenarios was the same. On the second experiment scenario the battery of the smartphone lasted 19 hours and 15 minutes, approximately.

Table V summarizes the third AMBRO mobile gateway experiment scenario.

Table V - AMBRO mobile gateway third experiment scenario.

Services	Tested Features					
	Activity Detection	*Check Heart Rate*	*Reading Frequency*	*Send Data to the Cloud*	*Smartwatch Battery*	*Smartphone Battery*
Location Monitoring Service	60 seconds	-------	-----	When a location update is requested	11h 15minutes	19h 30 minutes (approx.)
Heart Rate Monitoring Service	60 seconds	1 second	-----	10 minutes		
Fall Detection Service	-------	-------	100 milliseconds	When a fall is detected		

On the third experiment scenario the activity detection was performed every minute. The Shimmer module verified the acceleration and the angular velocity every 100 milliseconds. The information regarding the heart rate monitoring service was delivered to the cloud every 10 minutes. Those changes allowed the smartphone battery to last 19 hours and 30 minutes. (more 30 minutes than the first experiment scenario, and more 15 minutes than the second experiment scenario).

The variation of the parameters on the third experiment scenario allowed the smartphone battery to last more than on the other experiment scenarios. Nevertheless, the precision of the system services on that scenario was also lower. In the activity detection case, as the user activity was detected less often, it was more likely to user movements were not detected. Consequently, the number of location updates decreased, resulting on system performance degradation. On the fall detection system, as the acceleration and the angular velocity were not checked by the Shimmer module so often, the probability for a fall not being detected increased. This is due to the time interval on which a fall starts to be noticed being as small as possible, since a fall is originated by a sudden movement in the order of milliseconds.

The solution that was chosen to be implemented on the AMBRO mobile gateway application was the one described on first experiment scenario. This is due to the better commitment between the power consumption of the battery of the devices (of both smartwatch and smartphone) and the overall performance of the system, that took into account the activity detection, location updates, heart rate readings, sampling frequency in the Shimmer sensor module, and communication with the AMBRO platform.

7.2. Accuracy of the Services

The accuracy of the AMBRO mobile gateway services allow us to evaluate if the application is conducting his job properly. The location monitoring service accuracy can be obtained by evaluating the operating mode of the mechanisms used to detect the user activity and the GPS coordinates. The activity detection mechanism on the AMBRO mobile gateway application was used to detect if the user was quiet in some place or if he was in movement at a certain moment. If the user is still, and is not using his smartphone for any purpose, the activity detection mechanism always detects the user is still. If the user is walking or driving a car, the activity detection mechanism detects the user is in movement, and requests location updates through the most accurate location provider available at the moment (network, power cells or GPS). The accuracy of the location update mechanism depends on the location provider used to get the GPS coordinates. For example, if the location update mechanism is capable to use the GPS provider to retrieve the location coordinates, the accuracy ranges from 10 to 30 meters. However, if the location update mechanism is only capable to use the power cells to retrieve the location coordinates, the accuracy can drop to 2000 meters.

To test the accuracy of the location monitoring service, there were collected 1800 GPS coordinates samples when the user was outdoor and another equal sample when the user was indoor. Table VI shows the accuracy of the location monitoring service when the user was indoor and when the user was outdoor.

Table VI - Location monitoring service accuracy.

Location	Range (in meters)					
	<50m	*50-200m*	*200-500m*	*500-1000m*	*1000-2000m*	*>2000m*
Outdoor	91%	7%	2%	--------	--------	----------
Indoor	15%	15%	15%	17%	21%	17%

GPS coordinates are more accurate in outdoor locations with 91% of the gathered samples obtaining an accuracy over 50 meters. In comparison with indoor locations, only 15% of the samples obtained an accuracy better than 50 meters. These results were already expected, since at outdoor, locations the smartphone is more capable to get the location coordinates through the GPS system, while at indoor locations the coordinates are usually gathered through the network.

Regarding the heart rate monitoring service, the returned results are dependent on how accurate is the optical heart rate sensor embedded in the smartwatch. The accuracy of those types of sensors is good when the person who uses it is stopped, and deteriorates when the person is in movement. As those sensors sometimes return heartbeat values equal to zero (especially when the user is moving), the service cannot be used to detect if a person faints or collapses, since it could detect too many false positive values. Instead of that, the service evaluates the usual heartbeat when the person stops and the usual heartbeat when a person is moving. Figure 47 shows a graphic with the percentage of non-zero values when the smartwatch was used while the person stopped, and while the person was in movement. There were analyzed 18000 samples for each situation.

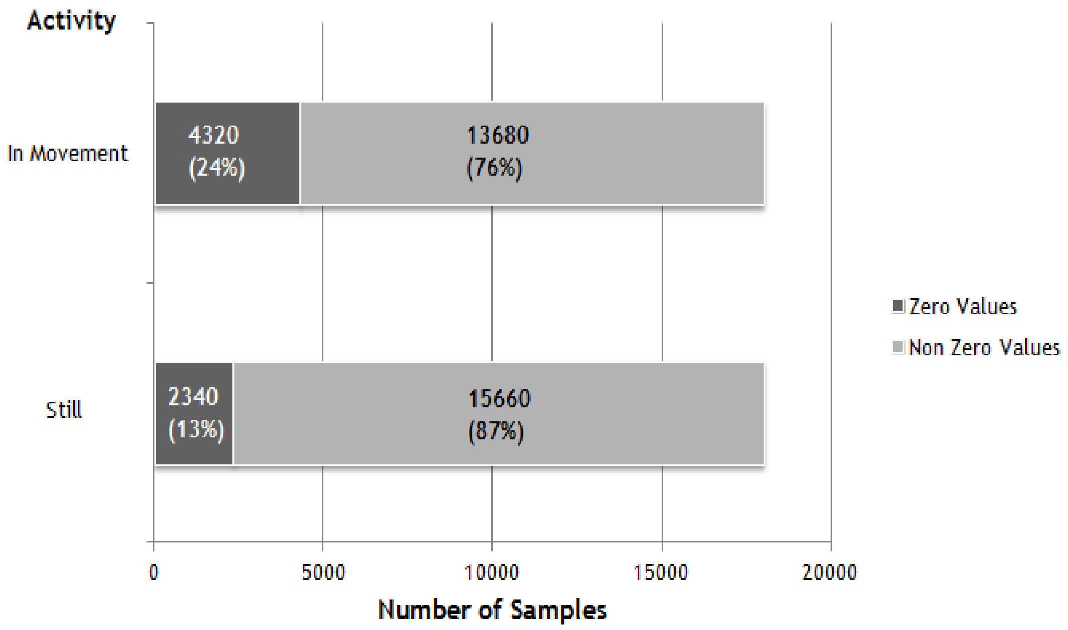

Fig. 47 Heart rate monitoring service accuracy.

Analyzing the results shown in Figure 47, while the user was stopped, the optical heart rate sensor returned values different than zero around 87% of the time. When the user was moving the results only shown valid values about 76% of the time. This can be related to the displacement of the smartwatch in the wrist of the user while he is moving.

The fall detection service must only detect a fall when a person effectively falls. But, sometimes when a person is performing some daily life activities, the algorithm used by the accelerometer and the gyroscope can fail, and detect false positives. Two scenarios where false positives can be originated happen when the user runs or jumps, since the thresholds used to detect falls can be sometimes surpassed. The fall detection service accuracy was tested by ascertaining how many falls were detected by the Shimmer module when the user performed different daily life activities. Table VII displays the results of the experiments performed to evaluate the accuracy of the fall detection service.

Table VII - Fall Detection service accuracy.

Activities Performed	Falls detected when performing different daily life activities		
	Number of tests	***Falls detected***	***Algorithm accuracy (%)***
Still	30	0	100%
Walk	30	0	100%
Sit down	30	0	100%
Stand up	30	0	100%
Lay down	30	0	100%
Jump	30	1	96.7%
Run	30	4	86.7%
Fall	30	28	93.3%

Through the analysis of the Table VII, it is possible to detect that when the user was running or jumping some falls were incorrectly detected. It can also be seen that during the experiments, when the person was truly falling, there were two times where no fall was detected, giving a 93.3% of accuracy in the fall detection algorithm.

7.3. Communication with other Objects

When the object search service is used, the AMBRO mobile gateway searches other objects in the boundaries through Bluetooth broadcast messages. This is a process that can cause a high battery drain and should if performed very often. Despite that, by using sensors from different vendors, like the embedded heart rate sensor from the Moto 360 smartwatch, the Shimmer sensors, and the embedded sensors from the smartphone that acts as gateway itself, the AMBRO mobile gateway uses a very important concept in IoT environments, the interoperability between objects [6][17]. To improve the overall performance of the system Bluetooth sesnors could be replaced by BLE sensors, since the BLE allows sensors to announce the services they should offer to other objects through a client-server model.

7.4. Summary

This chapter focused on the performance evaluation of the AMBRO mobile gateway application. In Section 7.1 were described the experiments performed to find a suitable solution for the design of the mobile gateway application, taking into account the relationship between the power consumption on the system devices and the overall performance of the services. Section 7.2 presented the experiments performed to determine the accuracy of each monitoring service. For the location monitoring service accuracy when the user was indoor and when the user was outdoor was analyzed. For the heart rate monitoring service, it was analyzed how the optical heart rate sensor embedded on the smartwatch behaved when the user was stopped and when the user was moving. If the user heart rate was different than zero, the value was considered valid, otherwise it was invalid. The accuracy of the fall detection service was evaluated by analyzing the service behavior while the user was performing different daily life activities. If the user fell, the service must detect it. If the service detects a fall when the user did not fell, an error occurred, and the service performance is affected. Section 7.3 presented a quick overview on the object search service importance for the AMBRO mobile gateway application, describing a possible strategy that could improve the service itself. Finally, this section summarizes the chapter.

8. Conclusions and Future Work

This chapter synthesizes the work presented along this dissertation, including the main achievements that result from this work. It also presents some suggestions that can be explored in further works about IPAs in IoT environments.

8.1. Conclusions

One objective of this dissertation was the review of the solutions that currently exist involving the insertion of IPAs on IoT scenarios. Therefore, a detailed study about the IoT technology was presented, including some protocols and technologies that can enable the dissemination of the IoT over the world. The main characteristics of IPA agents were also presented, including the features they need to exhibit to be inserted on ubiquitous communication environments. To complete all the research, there was developed a novel mobile gateway solution for a ubiquitous mobile health scenario, in which information gathered from a BSN was used by an IPA belonging to a caretaker.

The creation of the Internet introduced new forms of communication. Its evolution to create ubiquitous communication scenarios is the idea expressed by the IoT. The IoT will allow billions of devices to be connected on a common network. Different types of those devices and different demands can create some issues that need to be solved. One of those issues is related to the creation of a standard protocol to allow all devices to communicate among them by using low power mechanisms. Some protocols, as the XMPP, MQTT or CoAP are seen as suitable candidates, but a unique solution is yet to be created.

Currently, the human society uses technological gadgets on a regular basis. The creation of smartphones boosted this phenomenon. They include several features that humans can exploit to fulfill some of their daily life requirements. One of those features is the inclusion of personal assistants that are able to interact with their owners through voice and input commands. These personal assistant agents, or IPAs, are supposed to have a knowledge database about many subjects, including their owners' preferences. Thus, they make use of learning machine techniques and other AI disciplines. One strategy that can improve the assistance given by IPAs to their owners is the inclusion of these devices on IoT scenarios, since the number of information sources available is huge and can improve the knowledge database of IPAs.

AMBRO is an IPA platform that can be included on ubiquitous communication scenarios. The AMBRO personal assistant is supposed to help their owners on many of their daily life tasks, acting automatically on many situations. The architecture of the IPA platform defines the existence of several user interfaces (mobile devices, computers, smartwatches, Web enabled devices) that AMBRO users can use; a cloud infrastructure where many of the AMBRO platform processing is done (the generation of notifications for the user, storing of data, and creation of learning mechanisms for the IPA); and a external environment comprising several intelligent objects that AMBRO can interact with. In this last architecture layer, gateways to forward the traffic between the objects are needed. Static gateways can be used for static environments (smart home, for example). Mobile gateways are used for environments where objects are constantly moving (mobile health scenarios, for example).

The mobile gateway inserted on a mobile health scenario was the scenario explored for this dissertation. By using a Shimmer sensor module to detect falls, an optical heart rate sensor embedded on a smartwatch, and a GPS sensor included on a smartphone, a person is supposed to be monitored anywhere and anytime. This information is important to the AMBRO platform, since the person using the IPA can remotely check the location, the heart rate, or the information about possible falls of a person under his care at anytime.

The performance evaluation of the mobile gateway application considered important metrics for the IoT applications development. Initially, it was evaluated how the features of the application could vary to optimize the power consumption on the devices (smartphone and smartwatch) without degrading the system performance. It was difficult to perform that evaluation, but since the power consumption on the three experiment scenarios did not varied abruptly, the chosen solution was the one where the monitoring services gathered information more frequently. Other metric evaluated was related with the monitoring services accuracy. This evaluation was performed by assessing the accuracy of the samples gathered for each monitoring service. For the location monitoring service it was concluded that the GPS coordinates gathered by the smartphone were more accurate on outdoor location, as expected. For the heart rate monitoring service the accuracy of the heartbeat values measured by the optical heart rate sensor on the smartwatch was higher for scenarios where the user was stopped. For the fall detection service, the Shimmer device detected falls with high accuracy, with errors manly appearing for scenarios where the user was jumping or running. The interoperability between heterogeneous objects was also assured by putting heterogeneous objects, like the Motorola smartwatch, the Shimmer sensor module, and the Samsung smartphone communicating in the same environment.

By helping to explore more than one scenario, the AMBRO mobile gateway helped to prove that the AMBRO personal assistant can be used on different types of environments. This is an important point for the deployment of ubiquitous communication scenarios.

8.2. Future Work

To conclude this dissertation, some suggestions for future works are considered as follows:

- Develop the AMBRO mobile gateway application for other mobile operating systems, such as iOS;

- Include more objects on the mobile health scenario to gather more information about the health status of the monitored person;

- Deploy a scenario where all objects can communicate through BLE. By doing this, all objects on the environment can announce the services they have to offer, easing the communication between the mobile gateway and those same objects;

- Use objects that can communicate through low-power protocols, such as the CoAP;

- Develop new mechanisms that assure the communication between objects, mobile gateway, and the AMBRO platform is performed with less power.

References

[1] L. Kleinrock, “Information Flow in Large Communication Nets.” .

[2] B. M. Leiner, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, S. Wolff, and V. G. Cerf, “A Brief History of the Internet Professor of Computer Science,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22–31, 2009.

[3] R. Ebad, “Ubiquitous Computing : A Brief Review of Impacts and Issues,” 2011.

[4] D. Singh, G. Tripathi, and A. J. Jara, “A survey of Internet-of-Things: Future vision, architecture, challenges and services,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 287–292.

[5] J. J. P. C. Rodrigues and P. A. . C. S. Neves, “A survey on IP-based wireless sensor network solutions,” *Int. J. Commun. Syst.*, vol. 23, no. January, pp. 963–981, 2010.

[6] A. J. Jara, L. Ladid, and A. Skarmeta, “The Internet of Everything through IPv6 : An Analysis of Challenges , Solutions and Opportunities,” *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 4, no. 3, pp. 97–118, 2013.

[7] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, “Smart Home: Integrating Internet of Things with Web Services and Cloud Computing,” in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, 2013, pp. 317–320.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22-32, 2014.

[9] D. Singh, U. S. Tiwary, and W.-Y. Chung, “IP-based ubiquitous healthcare system,” in *2008 International Conference on Control, Automation and Systems*, 2008, pp. 131–136.

[10] P. Halapeti and S. Patil, “Healthcare Monitoring System Using Wireless Sensor Networks,” *Int. J. Adv. Res. Comput. Sci. Technol.*, vol. 2, no. 2, pp. 443–446, 2014.

[11] Z. Xiang-li, Y. Jin, Y. Kun, and L. Jian, "A remote manufacturing monitoring system based on the Internet of Things," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, 2012, pp. 221–224.

[12] L. Yuan, Y. Guo, J. Jiang, and L. Nian, "The Research on Monitoring of Discrete Manufacturing Process Based on Internet of Things," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 1186–1191.

[13] P. Garrido, F. J. Martinez, and C. Guetl, "Adding Semantic Web Knowledge to Intelligent Personal Assistant Agents," in *SWC 2010 - The Ninth International Semantic Web Conference*, 2010, pp. 1–12.

[14] "Ageing and life course, World Health Organization." [Online]. Available: http://www.who.int/ageing/en/.

[15] Y. P. Raiwani, "Internet of Things : A New Paradigm," *Int. J. Sci. Res. Publ.*, vol. 3, no. 4, pp. 1–4, 2013.

[16] C. Qiang, G. Quan, B. Yu, and L. Yang, "Research on Security Issues of the Internet of Things," *Int. J. Futur. Gener. Commun. Netw.*, vol. 6, no. 6, pp. 1–10, Dec. 2013.

[17] L. Tan and N. Wang, "Future internet: The Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, pp. 376–380.

[18] B. Katole, M. Sivapala, and V. Suresh, "Principle Elements and Framework of Internet of Things," *Int. J. Eng. Sci.*, vol. 3, no. 5, pp. 24–29, 2013.

[19] R. Klauck and M. Kirsche, "Chatty Things - Making the Internet of Things Readily Usable for the Masses with XMPP," in *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2012, pp. 60–69.

[20] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[21] H. Zhang and L. Zhu, "Internet of Things: Key technology, architecture and challenging problems," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, 2011, pp. 507–512.

[22] C. Xiang and X. Li, "General analysis on architecture and key technologies about Internet of Things," in *2012 IEEE International Conference on Computer Science and Automation Engineering*, 2012, pp. 325–328.

[23] M. Wu, T.-J. Lu, F. Ling, H. Du, and J. Sun, “Research on the architecture of Internet of Things,” in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, pp. 484–487.

[24] “ISA 100 Homepage.” [Online]. Available: http://www.isa100wci.org/.

[25] “WirelessHART Homepage.” [Online]. Available: http://www.hartcomm.org/index.html.

[26] “ZigBee Alliance Homepage.” [Online]. Available: http://www.zigbee.org/Products/CompliantPlatforms/ZigBeePROFeatureSet.aspx.

[27] “NXP Homepage.” [Online]. Available: http://www.nxp.com/documents/user_manual/JN-UG-3048.pdf.

[28] “ZigBee Alliance Homepage.” [Online]. Available: http://www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx.

[29] “Embedded.com Homepage.” [Online]. Available: http://www.embedded.com/design/connectivity/4419558/1/Zigbee-s-new-IP-specification-for-IPv6-6LoPAN-wireless-network-designs.

[30] “Jennic Homepage.” [Online]. Available: http://www.jennic.com/index.php.

[31] L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, “Routing and mobility approaches in IPv6 over LoWPAN mesh networks,” *Int. J. Commun. Syst.*, vol. 24, no. February, pp. 1445–1466, 2011.

[32] “IPSO Homepage.” [Online]. Available: http://www.ipso-alliance.org/wp-content/media/6lowpan.pdf.

[33] “Contiki OS Homepage.” [Online]. Available: http://www.contiki-os.org/.

[34] “Tiny OS Homepage.” [Online]. Available: http://www.tinyos.net/.

[35] L. M. L. Oliveira, J. J. P. C. Rodrigues, B. M. Mação, P. A. Nicolau, L. R. Wang, and L. Shu, “End-to-End Connectivity IPv6 over Wireless Sensor Networks,” in *2011 Third International Conference on Ubiquitous and Future Networks (ICUFN)*, 2011, pp. 1–6.

[36] B. Pediredla, K. I.-K. Wang, Z. Salcic, and A. Ivoghlian, “A 6LoWPAN implementation for memory constrained and power efficient wireless sensor nodes,” in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 4432–4437.

[37] T. Tsvetkov, “RPL : IPv6 Routing Protocol for Low Power and Lossy Networks,” in *Sensor nodes - Operation, Network and Application*, 2011, no. July, pp. 59–66.

[38] "cs.berkeley.edu." [Online]. Available: http://www.cs.berkeley.edu/~jwhui/6lowpan/IPSO-WP-7.pdf.

[39] E. Ancillotti, R. Bruno, and M. Conti, "On the interplay between RPL and address autoconfiguration protocols in LLNs," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013, pp. 1275–1282.

[40] J. Ko, D. Culler, S. Dawson-Haggerty, G. Omprakash, and A. Terzis, "Evaluating the Performance of RPL and 6LoWPAN in TinyOS," in *Proceedings of Extending the Internet to Low power and Lossy Networks*, 2011.

[41] C. Bormann, A. Castellani, and Z. Shelby, "CoAP : An Application Protocol for Billions of Tiny Internet Nodes," *Internet Comput. IEEE*, vol. 16, no. 2, pp. 62–67, 2012.

[42] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, "Evaluation of Constrained Application Protocol for Wireless Sensor Networks," in *2011 18th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN)*, 2011, no. 7, pp. 1–6.

[43] K. Kuladinithi, T. Pötsch, O. Bergmann, M. Becker, and C. Görg, "Implementation of CoAP and its Application in Transport Logistics," in *Proceedings of Extending the Internet to Low power and Lossy Networks*, 2011.

[44] J. Eliasson, J. Delsing, A. Raayatinezhad, and R. Kyusakov, "A SOA-based framework for integration of intelligent rock bolts with internet of things," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 1962–1967.

[45] "XMPP Standards Foundation Homepage." [Online]. Available: http://xmpp.org/xmpp-protocols/xmpp-extensions/.

[46] "XMPP Wiki." [Online]. Available: http://wiki.xmpp.org/web/Tech_pages/IoT_XepsExplained.

[47] A. Hornsby and E. Bail, "µXMPP: Lightweight implementation for low power operating system Contiki," in *2009 International Conference on Ultra Modern Telecommunications & Workshops*, 2009, pp. 1-5.

[48] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, and M. Ameling, "A Service Infrastructure for the Internet of Things based on XMPP," in *IEEE International Conference on Pervasive Computing and Communications*, 2013, no. March, pp. 385–388.

[49] Z. Shelby and C. Bormann, *6LoWPAN : The Wireless Embedded Internet*. 2009, p. 244.

[50] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks,” in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE ’08)*, 2008, pp. 791–798.

[51] S. Furrer, W. Schott, H. L. Truong, and B. Weiss, “The IBM Wireless Sensor Networking Testbed,” 2005.

[52] P. Maes, “Agents that reduce work and information overload,” *Commun. ACM*, vol. 37, no. 7, pp. 30–40, Jul. 1994.

[53] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. 2010, p. 682.

[54] S. B. Kotsiantis, “Supervised Machine Learning : A Review of Classification Techniques,” in *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007, vol. 31, pp. 3–24.

[55] Z. Ghahramani, “Unsupervised Learning,” in *Advanced Lectures on Machine Learning*, 2004, pp. 72–112.

[56] P. Stone and M. Veloso, “Multiagent Systems : A Survey from a Machine Learning Perspective,” *Auton. Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[57] G. Weiss, “Learning to Coordinate Actions in Multi-Agent Systems,” in *IJCAI’93 - Proceedings of the 13th international joint conference on Artifical intelligence - Volume 1*, 1993, pp. 311–316.

[58] “Apple Siri Webpage.” [Online]. Available: https://www.apple.com/ios/siri/.

[59] “Google Now Webpage.” [Online]. Available: http://www.google.com/landing/now/.

[60] “Samsung S Voice Webpage.” [Online]. Available: http://www.samsung.com/global/galaxys3/svoice.html.

[61] “LG Voice Mate Webpage.” [Online]. Available: http://www.lg.com/us/support/product-help/doc-1403913325190-en-LGEAI.

[62] “Microsoft Cortana Webpage.” [Online]. Available: http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana.

[63] B. H. Juang and L. R. Rabiner, “Automatic Speech Recognition – A Brief History of the Technology Development,” in *Elsevier Encyclopedia of Language and Linguistics*, 2005.

[64] C. Pozna, P. Foldesi, and J. Kovacs, “The Personal Assistant application, problem definition,” in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, 2013, pp. 851–856.

[65] B. Azvine, D. Djian, K. C. Tsui, and W. Wobcke, “The Intelligent Assistant: An Overview,” in *Intelligent Systems and Soft Computing*, 2000, pp. 215–238.

[66] M. Verlic, M. Mertik, and M. Zorman, “iAPERAS – Intelligent Athlete ’ s Personal Assistant,” in *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, 2005, vol. 386, no. 0, pp. 134–138.

[67] A. Komninos and S. Stamou, “HealthPal : An Intelligent Personal Medical Assistant for Supporting the Self-Monitoring of Healthcare in the Ageing Society,” in *4th International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2006.

[68] F. Li and N. Chen, “An intelligent assistant design system based on CBR and MAS,” in *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, 2009, pp. 1335–1339.

[69] E. Shaw, W. L. Johnson, and R. Ganeshan, “Pedagogical Agents on the Web,” in *Proceedings of the third annual conference on Autonomous Agents - AGENTS ’99*, 1999, pp. 283–290.

[70] B. El Saghir and N. Crespi, “An Intelligent Assistant for Context-Aware Adaptation of Personal Communications,” in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 2535–2540.

[71] E. C. Paraiso and J.-P. A. Barthès, “An Intelligent Speech Interface for Personal Assistants in R & D Projects,” in *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design*, 2005, pp. 804–809.

[72] “Protégé Webpage.” [Online]. Available: http://protege.stanford.edu/.

[73] M. Fuckner, J.-P. Barthès, and E. E. Scalabrin, “Using a personal assistant for exploiting service interfaces,” in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2014, pp. 89–94.

[74] G. Czibula, A.-M. Guran, I. G. Czibula, and G. S. Cojocar, “IPA - An intelligent personal assistant agent for task performance support,” in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 31–34.

[75] G. Rao. S, Z. Suryady, U. Sarwar, and M. Abbas, “A gateway solution for IPv6 wireless sensor networks,” in *2009 International Conference on Ultra Modern Telecommunications & Workshops*, 2009, pp. 1–6.

[76] B. Silva, J. J. P. C. Rodrigues, L. M. L. Oliveira, L. D. P. Mendes, E. F. Nakamura, and C. M. S. Figueiredo, “Design and Construction of a Wireless Sensor and Actuator Network Gateway based on 6LoWPAN,” in *EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE*, 2011, pp. 1–4.

[77] B. Silva, J. J. P. C. Rodrigues, L. D. P. Mendes, E. F. Nakamura, and C. M. S. Figueiredo, “Design and Construction of Wireless Sensor Network Gateway with IPv4 / IPv6 Support,” in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[78] L. F. Schrickte, C. Montez, R. De Oliveira, and A. R. Pinto, “Integration of Wireless Sensor Networks to the Internet of Things Using a 6LoWPAN Gateway,” in *2013 III Brazilian Symposium on Computing Systems Engineering*, 2013, pp. 119–124.

[79] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “On the Integration of Cloud Computing and Internet of Things,” in *2014 International Conference on Future Internet of Things and Cloud (FiCloud)*, 2014, pp. 23–30.

[80] J. Zhou, T. Leppänen, E. Harjula, C. Yu, H. Jin, and L. T. Yang, “CloudThings : a Common Architecture for Integrating the Internet of Things with Cloud Computing,” in *Proceedings of the 2013 IEEE 17th Internetional Conference on Computer Supported Cooperative Work in Design*, 2013, pp. 651–657.

[81] C. Dores, L. P. Reis, and N. V. Lopes, “Internet of things and cloud computing,” in *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 2014, pp. 1–4.

[82] P. R. B. B., P. Saluja, N. Sharma, A. Mittal, and S. V. Sharma, “Cloud Computing for Internet of Things & Sensing Based Applications,” in *2012 Sixth International Conference on Sensing Technology*, 2012, pp. 374–380.

[83] R. Roman, P. Najera, and J. Lopez, “Securing the Internet of Things,” *IEEE Comput.*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[84] S. L. Keoh, S. S. Kumar, and H. Tschofenig, “Securing the Internet of Things: A Standardization Perspective,” *IEEE Internet Things J.*, vol. 1, no. 3, pp. 265–275, Jun. 2014.

[85] A. P. Abidoye, N. A. Azeez, A. O. Adesina, K. K. Agbele, and H. O. Nyongesa, “Using Wearable Sensors for Remote Healthcare Monitoring System,” *J. Sens. Technol.*, vol. 01, no. 02, pp. 22–28, 2011.

[86] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, “An Advanced Wireless Sensor Network for Health Monitoring,” in *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, 2006, pp. 2–5.

[87] “AlertMe Webpage.” [Online]. Available: http://www.alertme.com/.

[88] “Lowes Webpage.” [Online]. Available: http://www.lowes.com/cd_What+Is+Iris_695688710_?cm_sp=Electrical-_-HomeAutomation|HB-_-Merch|Iris_Search_Learn_More.

[89] S. Hwang and D. Yu, “Remote Monitoring and Controlling System Based on ZigBee Networks,” *Int. J. Softw. Eng. Its Appl.*, vol. 6, no. 3, pp. 35–42, 2012.

[90] S. Kumar, “Ubiquitous Smart Home System Using Android Application,” *Int. J. Comput. Networks Commun.*, vol. 6, no. 1, pp. 33–43, 2014.

[91] W. Fu, G. Chen, P. Wang, Y. Hong, and H. Ge, “Improvement of Home Appliance Control System in Smart Home Based on 6LoWPAN,” *J. Comput.*, vol. 9, no. 1, pp. 183–188, Jan. 2014.

[92] P. Santos, L. Varandas, T. Alves, C. Romeiro, J. Casal, S. Lourenço, and J. Santos, “A Pervasive System Architecture for Smart Environments in Internet of Things Context,” in *ICMI 2015 : XIII International Conference on Multimodal Interaction, London, UK*, 2015.

[93] “Z-Wave Alliance Homepage.” [Online]. Available: http://z-wavealliance.org/.

[94] S. Vanjire, U. Kanchan, G. Shitole, and P. Patil, “Location Based Services on Smart Phone through the Android Application,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 1, pp. 4982–4987, 2014.

[95] A. Chavan, S. Kamble, A. Bhagyshree, R. Sonawane, and R. C. Gadri, “Land Vehicle Tracking System Using Java on Android Platform,” *Comput. Eng. Intell. Syst.*, vol. 3, no. 5, pp. 88–94, 2012.

[96] A. Al-Mazloum, E. Omer, and M. F. A. Abdullah, “GPS and SMS-Based Child Tracking System Using Smart Phone,” *Int. J. Electr. Comput. Electron. Commun. Eng.*, vol. 7, no. 2, pp. 171–174, 2013.

[97] R. Kinage, J. Kumari, P. Zalke, and M. Kulkarni, “Mobile Tracking Application,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 2, no. 3, pp. 617–623, 2013.

[98] Q. T. Huynh, U. D. Nguyen, S. V Tran, A. Nabili, and B. Q. Tran, “Fall Detection System Using Combination Accelerometer and Gyroscope,” in *Proceedings of the Second International Confernece on Advances in Electronic Devices and Circuits - EDC 2013*, 2013, vol. 3200, pp. 978–981.

[99] S. Hwang, M. Ryu, Y. Yang, and N. Lee, “Fall Detection with Three-Axis Accelerometer and Magnetometer in a Smartphone,” in *International Conference on Computer Science and Technology , CST 2012*, 2012, pp. 65–70.

[100] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang, “A Wearable Wireless Fall Detection System with Accelerators,” in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, no. 2, pp. 2259–2263.

[101] “Android Studio Homepage.” [Online]. Available: https://developer.android.com/sdk/index.html.

[102] “Android Wear Wearable Message API.” [Online]. Available: https://developer.android.com/training/wearables/data-layer/messages.html.

[103] “Activity Recognition API Webpage.” [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi.

[104] “Google Cloud Messaging Webpage.” [Online]. Available: https://developers.google.com/cloud-messaging/.

Appendix

On this appendix are included the papers containing the main contributions of this dissertation. The first paper is entitled “Internet of Things Mobile Gateway Services for Intelligent Personal Assistants”, and presents a novel mobile gateway solution for ubiquitous mobile health scenarios. It was submitted to the IEEE HEALTHCOM on the 15th May 2015.

The second paper is entitled “Intelligent Personal Assistants Based on Internet of Things Approaches” and was submitted to the IEEE Systems Journal by August 28th, 2015, and it is under review.

The third paper is entitled “A IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments”. It was submitted to the Journal of Biomedical Informatics in October 2015, and it is under review.

Internet of Things Mobile Gateway Services for Intelligent Personal Assistants

João Santos[1], Bruno M. C. Silva[1], Joel J. P. C. Rodrigues[1,2], João Casal[3], Kashif Saleem[4]

[1] Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
[2] University of Fortaleza (UNIFOR), Ceará, Brazil
[3] TIMWE I &D, Parkurbis – Parque de Ciência e Tecnologia da Covilhã, Tortosendo, Portugal
[4] Center of Excellence in Information Assurance (CoEIA), King Saud University (KSU), Riyadh, Kingdom of Saudi Arabia

joao.santos@it.ubi.pt; bruno.silva@it.ubi.pt; joeljr@ieee.org; joao.casal@timwe.com; ksaleem@ksu.edu.sa

***Abstract*— The wide dissemination of Internet around the world changed the way people communicate among them. The evolution of the information and communication technologies (ICT) or artificial intelligence (AI) allowed the creation of new types of services involving electronic devices with huge potential. The emergence of intelligent personal assistants (IPAs) is one of such potential example. Combining the IPA concept with the recent paradigm of Internet of Things (IoT), offers new possibilities and new services to end users, since they could learn more about other entities present in the surrounding environment. This paper proposes a novel mobile gateway solution for a ubiquitous mobile health scenario, in which information gathered from a body sensor network (BSN) is used by an IPA belonging to another person, typically mentioned to as caretaker. The mobile gateway receives real time information related to location, heart rate, and possible falls of the monitored person and acts as a communication channel between the BSN sensors and the IPA platform. Furthermore, the paper presents a performance evaluation study of the proposed mobile IoT-based gateway demonstrating and validating its feasibility.**

Keywords*— *Internet of Things; Mobile Gateway; Intelligent Personal Assistant; Body Sensor Network

I. Introduction

The development of several technological topics offer humans the ability to create services capable of giving them serious advantages for their daily lives. Some of the technological fields that can be enumerated are related with the artificial intelligence (AI), such as speech recognition, natural language processing, semantic Web, or machine learning [1]. While others are related to electronics and information and communication technologies [2, 3]. The exploration of all these topics combined with all the information that Internet turns accessible created opportunities to create smart devices capable to gather data and "understand" the environment that surrounds them. These devices are called Intelligent Personal Assistants (IPAs) [4-7].

The introduction of IPA agents in the society can automate and expedite the fulfilling of many daily life tasks. The assistance that IPAs give to their owners improves more autonomous and context aware solutions. The autonomy and context awareness of the IPAs are higher if they are capable to communicate with other objects available in their communication environment, since they have more information sources to interact, increasing their learning database. The establishment of the communication between every addressable object like IPAs and other objects in the environment can be enabled by the IoT paradigm [8, 9]. The integration of IPAs in IoT scenarios allows the improvement of certain services, like those related to healthcare, ambient assisted living, sports, among others.

This paper presents a novel mobile IoT gateway solution that is integrated on an IPA platform, called AMBRO. A body sensor network (BSN) is created to monitor the behavior of a single person, gathering information about the location, heart rate and possible falls of the user to an IPA that belongs to a caretaker [10]. All this information is forwarded to the IPA platform through the mobile gateway that is always with the person under monitoring. Then, the person is always under monitoring, creating a ubiquitous mobile health scenario. Besides that, it is also evaluated how the mobile gateway performs in this scenario of ubiquitous communication, essentially in terms of power consumption. The main contributions of this paper include the creation of a novel mobile gateway solution to integrate with an IPA on a IoT scenario; a performance assessment study of AMBRO platform that proves its feasibility under multiple environments; and a novel remote monitoring system integrated on a mobile health scenario.

The remainder of this paper is organized as follows. Section II reviews the related literature about IPAs, remote monitoring systems, and IoT gateways. Section III shows the different layers of the AMBRO platform, and Section IV describes the characteristics of the AMBRO mobile gateway application, indicating the services and their corresponding capabilities. Section V discusses the performed experiments in the application and the obtained results, while Section VI concludes the paper and addresses some future work.

II. Related Work

In recent years, several works have been presented involving IPAs where some of them are focused on healthcare. In [11], the authors proposed a mobile health monitoring system which monitors the health condition of the elderly people through the use of peripheral devices, such as thermometers and blood pressure gauges. All those indicators are transferred to PDA devices of the physicians, updating the information about the condition of their patients. The authors of [12] propose a network architecture for continuous monitoring of patients. Using wireless sensor networks (WSNs), it aims to create a system where various heterogeneous devices help to monitor patients living inside a nursing home, in real time. All the gathered information about the monitored people is transmitted to a backend database, and it is accessible through a PDA or a PC of the caretakers. In [13] is shown a system for remote healthcare monitoring. It uses a wireless body area network (WBAN) to gather health information about the patients. The system considers three tiers, where the first represents the body sensors that are attached to the patients, the second tier corresponds to the intelligent personal digital assistant (IPDA) of the monitored people that acts as the system gateway, while the third tier is the system backbone where data is saved for future reference by a medical staff. Still in the healthcare area, in [14] is shown a technical approach for developing a healthcare system based on a 6LoWPAN network, where the body sensors retrieve information about their vital signs, sending it directly to a mote gateway connected to the Internet. In this solution, physicians are able to observe the condition of their patients, in real time, using their PDAs with access to the Internet.

The construction of smart gateways for IoT environments is a case study for the scientific community. The available technology allows the creation of dedicated gateways, like the Power over Ethernet (PoE) gateway presented in [15]. There, it is demonstrated how a stationary gateway with capability to communicate with healthcare sensors over Bluetooth, Wi-Fi or IEEE 802.15.4 protocols can be designed. The authors of [16] present a mobile gateway architecture for IoT. The mobile gateway, typically a smartphone or a PDA, was designed to provide a suitable solution for ubiquitous healthcare systems, where the patients are supposed to be monitored continuously. Using Bluetooth to retrieve data from sensors, the mobile gateway sends that information for the medical staff analysis. In [17], authors study the major problems that can rise when developing a smartphone mobile gateway for IoT environments, such as the security and integrity of the sensor data, as well as the correct identification of the sensors in the environment. In addition, it is presented an Android gateway prototype that collect data from temperature sensors, which is then sent to some network applications to consume. In [18], it is presented a wireless gateway for Machine-to-Machine (M2M) applications. The proposed gateway is capable to make the logical connection between the sensor and smart object layer and the client layer. The client interface of the gateway is able to dynamically discover M2M objects, notifying the clients when a new object is found, allowing those clients to retrieve data gathered from those registered devices. Authors of [19] present a ubiquitous health monitoring system using a Android smartphone as the system gateway. The sensors that monitor the vital signs of the persons communicate with a sink node through the ZigBee protocol. Then, the sink node connects to an USB port of the Android smartphone, sending by this way, the data retrieved from the BSN to the gateway. The gateway forwards the data from the BSN to a server over the Internet, where all the data is analyzed. In [20], the authors investigate some of the problems that still available to develop a universal solution for the interconnectivity between WSNs and Internet hosts, proposing a smartphone Bluetooth Low Energy (BLE) gateway to connect the IoT peripherals with the Internet.

The mobile gateway proposed in this paper focuses on a specific application for healthcare scenarios where persons are monitored in real time (location, heart rate or a possible fall detection). When compared with other available solutions, this mobile IoT-based gateway differs from the above surveyed solutions by having the particularity of being a communication channel and an object with decision-making capabilities at the same time. These characteristics transform the presented proposal into a device both able to connect the BSN to the Internet and a device capable of using the monitoring services with low power consumption while preserves the services accuracy. This gateway is designed for use case scenarios where mobility support may be needed, creating a challenging scenario.

III. The AMBRO Platform

The AMBRO platform [21] considers the proposal and creation of an IPA agent able to both interact with its user and other objects in the environment, creating a scenario of ubiquitous communication. As shown in Figure 1, the AMBRO platform is based on three main layers, *i)* IoT "world"; *ii)* intelligent cloud systems; and *iii)* user interfaces.

The IoT world layer includes all the external entities (smart objects, sensors, services) that can give IPAs some useful information for their learning process, but they are not directly obtained from the behavior of their users. The Intelligent cloud system layer works like "the brain" of the whole system, where all the IPA logical actions occur. Databases, system intelligence, voice recognition, security and notification generator modules are all presented at this layer. The User Interface layer provides the AMBRO user interfaces, such as the back-office tool to manage objects and sensors, and the IPAs' interfaces. Then, they can be emphasized the IPA Web accessible in every device with Internet connectivity and the IPA mobile created for Android OS.

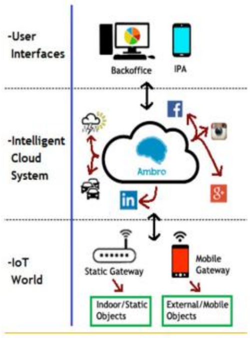

Fig. 1. AMBRO main architecture.

Analyzing the connection between the IoT world and the Intelligent Cloud System layers, it can be seen that, in some cases, there are needed gateways to transfer the traffic between both layers. For indoor and static environments, like smart homes, there are used static gateways. For environments in which devices (as sensors and smart objects) are mobile, it is necessary to use a mobile gateway. A possible solution to develop a mobile IoT gateway is the use of a smartphone. This paper presents a solution for the mobile gateway.

IV. AMBRO MOBILE GATEWAY

The AMBRO personal assistant is designed to be a tool to support people to perform the most variety of daily tasks. Then, AMBRO must be capable to: *i*) let people control their home environment by controlling smart objects distributed around the house; *ii*) give useful information about external events through Web services (e.g., weather, traffic); *iii*) generate useful suggestions for the user taking into account his previous behavior; and *iv*) remotely monitor a person that is under the care of another person/user. This last use-case is explored in this paper, proving that AMBRO can be used in different environments. Considering this scenario (*iv*), there are needed two entities: the caretaker and the monitored person. The caretaker will have the AMBRO personal assistant application installed on his smartphone, and the monitored person is monitored by sensors attached to him, on a BSN scenario. The monitored person always needs a smartphone with the AMBRO mobile gateway application installed to run the monitoring services.

The AMBRO mobile gateway application was developed for Android OS. To use it, the user first needs to login the application with his AMBRO account. After correct login, the main screen shows up, presenting the option to start each individual service (location monitoring, heart rate monitoring, and fall detection), search Bluetooth objects to detect sensors used on the fall detection service, and the logout option. Figure 2 shows the AMBRO mobile gateway main screen user interface.

Fig. 2. AMBRO mobile gateway main screen user interface.

The AMBRO mobile gateway architecture comprises the four following modules: the BSN, the mobile gateway, the AMBRO cloud, and the IPA of the caretaker.

Figure 3 depicts the AMBRO mobile gateway system architecture.

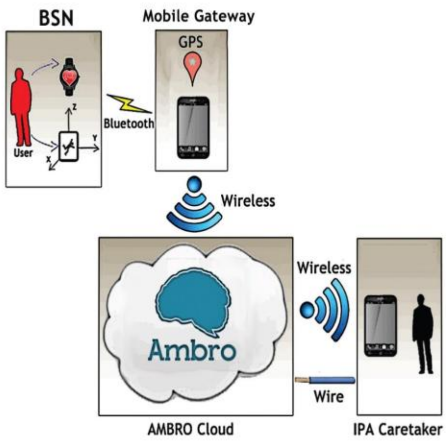

Fig. 3. Illustration of the AMBRO mobile gateway system architecture.

Each service of the AMBRO mobile gateway application is associated to a sensor that is gathering physiological data from the person under monitoring. The location monitoring service is used to detect the current location of the person using the GPS system of the smartphone that acts as gateway. The heart rate monitoring service uses a smartwatch with an embedded heart rate sensor to count the user heartbeats. The fall detection service uses an accelerometer/gyroscope module to detect user falls. Both heart rate and accelerometer/ gyroscope modules send data to the mobile gateway through a Bluetooth connection. After, the gateway sends the corresponding data to each service to AMBRO cloud via a wireless Internet connection. Inside AMBRO cloud, the received data is stored and analyzed, and if some value coming from the AMBRO mobile gateway is not inside the range that it was expected, an alert notification is generated to the caretaker IPA.

A. Location monitoring service

The location monitoring service available at the AMBRO mobile gateway is used to check the position of a person under monitoring using the GPS of the smartphone. Using the low power embedded sensors of the smartphone, the location monitoring service is able to detect the activity that a person is currently performing, only requesting location updates when the person is moving.

When the location service is initiated, activity updates are requested at each 20 seconds. If the smartphone detects that the user is moving, GPS is turned on and location updates are requested. If the smartphone detects that user still during one complete minute, location updates stop detection. This fact is useful to diminish the power consumption of the smartphone. If the data is not delivered to the AMBRO cloud, it is saved on the smartphone to be forwarded later. If the coordinates that are delivered to the AMBRO cloud (by the mobile gateway) return a place that is outside the area that was dynamically defined by the AI module (available in the cloud system), an alert notification is generated to the IPA of the caretaker.

B. Heart rate monitoring service

For the development of the heart rate monitor service it was used a smartwatch to constantly measure the heart rate of the person under monitoring. Some wearable devices like the Moto 360 used in this project, uses the Android Wear operating system, allowing programmers to make applications for wearable devices in a similar process to the Android smartphone applications development. The Moto 360 reads the heart rate of the monitored person every second, sending that information to the smartphone through a Bluetooth connection. The smartphone detects the activity that its user is performing at 20 seconds rate. This information is useful to detect if the heart rate collected values are normal or not, depending on the activity of the monitored person. That evaluation is performed inside the AMBRO platform by the AI module after the heart rate and activity information has been sent from the AMBRO mobile gateway.

C. Fall detection service

One of the possibilities to detect a person fall is based on the use of tri-axis accelerometers. Attaching an accelerometer to a person body, the system is able to detect anytime and anywhere a possible fall. To improve the fall detection systems, there can also be used gyroscopes to detect the orientation of a person by calculating the angular velocity. An accelerometer and a gyroscope are used on the fall detection service of the AMBRO mobile gateway. Both acceleration and angular velocity values are compared with thresholds that are defined by the person who designs the system. There exist several examples of fall detection systems using only accelerometers like the work published by the authors in [22], or the combination of both accelerometers and gyroscopes available at [23, 24].

For the purpose of this work it was chosen the Shimmer Sensor Platform, since it is a wireless sensor platform very useful to develop applications for different purposes. Shimmer Sensor Platform includes biophysical, ambient and kinematic sensors, being this last type of sensors the ones used in the fall detection service of the AMBRO mobile gateway application.

D. Object search service

The AMBRO mobile gateway application allows the discovery of sensors that communicate over Bluetooth in the environment, such as the Shimmer sensors. Then, the user can select different sensors to be used in the fall detection service. To discover Bluetooth devices, the gateway sends a broadcast. Active Bluetooth devices in the environment can answer positively to this broadcast message. When the discovery process is over, a list of active Bluetooth devices is returned to the user. Then, the user can select one of those devices to be paired with the gateway. When the user launches the fall detection service he/she can choose the sensor to be used in that service.

V. RESULTS ANALYSIS

The performance evaluation of the AMBRO mobile gateway included three main metrics: *i*) power consumption; *ii*) accuracy of the results returned by the different services; and *iii*) communication with objects.

A. Power consumption

In an IoT environment, the power consumption is very important for the overall performance of the system since each device is usually powered by batteries and it is supposed to be able to communicate anytime and anywhere. To avoid the quick discharge of the batteries it is necessary to make devices communicate only when needed. In the AMBRO mobile gateway application, different experiments were conducted to evaluate the parameters that could change in order to optimize the power consumption of the system. For both location monitoring and heart rate monitoring services, experiments were conducted to detect the activity that the user was performing on periodic updates of 20, 40, and 60 seconds. On the heart rate monitor service there was also changed the periodicity in which the mean values of the user heartbeat were sent to the cloud, with the periodicity for 5 to 10 minutes. On the fall detection service, the reading frequency of both acceleration and angular velocity values of the Shimmer module changed between 20, 50, and 100 milliseconds. The smartphone used for these experiments was the Samsung Galaxy Express 2, equipped with the Android OS 4.2.2, CPU dual-core 1.7 GHz, internal memory up to 8 GB and 1.5 GB RAM, while the smartwatch used was the Moto 360, equipped with the Android Wear OS (version 5.0.2), a Texas Instruments OMAP 3 processor, 4 GB of internal memory and 512 MB RAM. During the experiments, no other applications and services were running. It should be noted that both smartphone and smartwatch batteries can handle, at least, 40 hours when used on normal daily user activities. Table I shows the solution that was used by the AMBRO mobile gateway application, as well as the battery life-time of both smartwatch and smartphone when used that same solution.

TABLE I. POWER CONSUMPTION IN THE SMARTPHONE AND IN THE SMARTWATCH.

Services	Tested Features					
	Activity Detection	***Check HR***	***Reading Frequency***	***Send Data to the Cloud***	***Smartwatch Battery***	***Smartphone battery***
Location Service	20 sec	------	-----	When a location update is requested	11h 15mins	19h (approx.)
HR Service	20 sec	1 sec	-----	5 minutes		
Fall Detection Service	------	-------	20 msec	When a fall is detected		

The chosen solution was the better compromised with the battery power consumption (of both smartwatch and smartphone) and the overall performance of the system, taking into account the activity detection, the location updates, the heart rate readings, the sampling frequency in the Shimmer sensor, and the communication with the AMBRO platform made more frequently.

B. Accuracy of the services

The accuracy of the location monitoring service can be measured evaluating the accuracy of the activity detection and the GPS location update mechanisms. The activity detection mechanism was used in the AMBRO mobile gateway application with the objective to detect the user movements (stopped or moving). If the user is walking or driving a car, the activity detection mechanism considers the user under movement, requesting location updates through the most accurate location provider available at the moment (network, power cells, or GPS). The accuracy of the location update mechanism is dependent of the location provider that is used for retrieving the GPS coordinates. For example, if the location update mechanism is capable to use the GPS provider to retrieve the location coordinates, the accuracy ranges from 10 to 30 meters. However, if the location update mechanism is only capable to use the power cells to retrieve the location coordinates, the accuracy can drop to 2000 meters.

Table II shows some results regarding the accuracy of the GPS used in the location monitor service with the user indoor and when the user was outdoor. There were collected 1800 GPS coordinates samples when the user was outdoor and another equal sample when the user was indoor. As expected, the GPS accuracy is higher in outdoor locations. Most of the gathered samples obtained an accuracy over 50 meters (91%). In comparison with indoor locations, only 15% of the samples obtained an accuracy better than 50 meters.

TABLE II. ACCURACY OF THE GPS LOCATION MONITOR SERVICE.

Location	Range (in meters)					
	<50m	*50-200m*	*200-500m*	*500-1000m*	*1000-2000m*	*>2000m*
Outdoor	91%	7%	2%	--------	--------	------
Indoor	15%	15%	15%	17%	21%	17%

Regarding the heart rate monitoring service, the returned results are dependent on how accurate is the optical heart rate sensor embedded in the smartwatch. As those sensors sometimes returns heartbeat values equal to zero (especially when the user is moving), the service cannot be used to detect if a person faints or collapses, since it could detect too many false positive values. Instead of that, the service evaluates the usual heartbeat when the person stops and the usual heartbeat when a person is moving. Table III shows the percentage of non-zero values when the smartwatch was used while the person stopped and when the person was in movement.

TABLE III. ACCURACY OF THE HEART RATE MONITOR SERVICE.

Activity	HR Samples and respective accuracy			
	Number of Samples	*Zero Values*	*Non Zero Values*	*Accuracy (%)*
Still	18000	2340	15660	87%
In Movement	18000	4320	13680	76%

Analyzing the results shown in Table III, it can be seen that, in the experiments performed while the user was stopped, the optical heart rate sensor returned values different to zero around 87% of the time, while when the user was under movement the results only shown valid values about 76% of the time. This can be related to the displacement of the smartwatch in the wrist of the user while he/she is moving.

The fall detection service must only detect a fall when a person effectively falls. However, sometimes, when a person is performing some daily life activities, the algorithm used by the accelerometer and the gyroscope can fail, and detect false positives. Table IV shows the results of the experiments performed to evaluate the precision of the fall detection service, with the Lower Acceleration Threshold (LAT) being 0.3 G, the Upper Acceleration Threshold (UAT) being 2.5G, and the Angular Velocity Threshold (AVT) being 220°.

TABLE IV. ACCURACY OF THE FALL DETECTION SERVICE.

Activities Performed	Falls detected when performing different daily life activities		
	Number of tests	*Falls detected*	*Algorithm accuracy(%)*
Still	30	0	100%
Walk	30	0	100%
Sit down	30	0	100%
Stand up	30	0	100%
Lay down	30	0	100%
Jump	30	1	96.7%
Run	30	4	86.7%
Fall	30	28	93.3%

Analyzing the Table IV, it is possible to detect that when the user was running or jumping some falls were incorrectly detected. It can also be seen that during the experiments, when the person was truly falling, there were two times where no fall was detected, giving a 93.3% of accuracy in the fall detection algorithm.

C. Communication with other objects

The AMBRO mobile gateway is able to search other objects in the boundaries through Bluetooth broadcast messages (objects discovery). This is a process that can cause a high battery drain if performed very often. Despite that, by using sensors from different vendors, like the embedded heart rate sensor from the Moto 360 smartwatch, the Shimmer sensors, and the embedded sensors from the smartphone that acts as gateway itself, the AMBRO mobile gateway uses a very important concept in IoT environments, the interoperability between objects [25, 26].

VI. Conclusion and Future Work

This paper presented a novel IoT-based mobile gateway that receives real-time data from multiple sensors inside a BSN. Each sensor or set of sensors can collect data that will be used to evaluate the condition of a person under monitoring (such as location, heart rate, etc.). A performance evaluation study of this solution was performed, proving its feasibility. The proposed services were successfully validated considering the GPS accuracy and power consumption when the respective services were in use.

For future work, it is considered the use of body sensors using Bluetooth Low Energy technology, since it would allow the creation of a scenario where the power consumption is even lower and where the sensors would possibly announce the services they should offer to the gateway. Another aspect to be considered is the one related with the intelligence of the system, deciding which or whom network entities are responsible for taking decisions about network traffic forwarding. Also, other services and applications can be enabled on the devices to better evaluate the power consumption when the monitoring services are running.

Acknowledgments

This work has been partially supported by TIMWE *Investigação e Desenvolvimento*, through the AMBRO project, co-funded by COMPETE/QREN/EU, in the context of the *Portuguese Sistema de Incentivos à I&DT* (project 33779), by the National Plan of Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (12-INF2817-02), by Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project.

References

[1] G. Chollet, J. Boudy, A. Esposito, G. Pelosi, and I. Mines-t, “Building The Next Generation of Personal Digital Assistants,” in *1st International Conference on Advanced Technologies for Signal and Image Processing - ATSIP'2014*, 2014, pp. 458–463.

[2] N. Koshizuka and K. Sakamura, “Ubiquitous ID: Standards for Ubiquitous Computing and the Internet of Things,” pp. 98–101, 2010.

[3] X. Jia, Q. Feng, T. Fan, and Q. Lei, “RFID technology and its applications in Internet of Things (IoT),” *2012 2nd Int. Conf. Consum. Electron. Commun. Networks*, pp. 1282–1285, Apr. 2012.

[4] G. Czibula, A.-M. Guran, I. G. Czibula, and G. S. Cojocar, “IPA - An intelligent personal assistant agent for task performance support,” *2009 IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, pp. 31–34, Aug. 2009.

[5] C. Pozna, P. Foldesi, and J. Kovacs, “The Personal Assistant application, problem definition,” *2013 IEEE 4th Int. Conf. Cogn. Infocommunications*, pp. 851–856, Dec. 2013.

[6] B. El Saghir and N. Crespi, “An Intelligent Assistant for Context-Aware Adaptation of Personal Communications,” *2007 IEEE Wirel. Commun. Netw. Conf.*, pp. 2535–2540, 2007.

[7] M. Fuckner, J.-P. Barthes, and E. E. Scalabrin, “Using a personal assistant for exploiting service interfaces,” *Proc. 2014 IEEE 18th Int. Conf. Comput. Support. Coop. Work Des.*, pp. 89–94, May 2014.

[8] H. Pu, J. Lin, F. Liu, and L. Cui, “An intelligent interaction system architecture of the internet of things based on context,” *5th Int. Conf. Pervasive Comput. Appl.*, pp. 402–405, Dec. 2010.

[9] K. A. Hribernik, Z. Ghrairi, C. Hans, and K. Thoben, “Co-creating the Internet of Things - First Experiences in the Participatory Design of Intelligent Products with Arduino,” in *Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011)*, 2011, no. Ice, pp. 1–9.

[10] B. M. Silva, T. Simões, J. J. P. C. Rodrigues, S. Sendra, and J. Lloret, “An Ambient Assisted Living Framework for Mobile Environments,” in *IEEE-EMBS International Conferences on Biomedical and Health Informatics (BHI 2014), Valencia, Spain*, 2014, pp. 448 – 451.

[11] A. Komninos and S. Stamou, “HealthPal : An Intelligent Personal Medical Assistant for Supporting the Self-Monitoring of Healthcare in the Ageing Society.”

[12] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, “An Advanced Wireless Sensor Network for Health Monitoring.”

[13] A. P. Abidoye, “Using Wearable Sensors for Remote Healthcare Monitoring System,” *J. Sens. Technol.*, vol. 01, no. 02, pp. 22–28, 2011.

[14] D. Singh and U. S. Tiwary, “IP-based ubiquitous healthcare system,” *2008 Int. Conf. Control. Autom. Syst.*, pp. 131–136, Oct. 2008.

[15] J. Granados, A.-M. Rahmani, P. Nikander, P. Liljeberg, and H. Tenhunen, “Towards Energy-Efficient HealthCare: an Internet-of-Things Architecture Using Intelligent Gateways,” *Proc. 4th Int. Conf. Wirel. Mob. Commun. Healthc. - "Transforming Healthc. through Innov. Mob. Wirel. Technol.*, 2014.

[16] K. Lee, Y. E. Gelogo, and S. Lee, “Mobile gateway System for Ubiquitous system and Internet of Things Application,” *Int. J. Smart Home*, vol. 8, no. 5, pp. 279–286, 2014.

[17] J. Li, Y. Zhang, Y.-F. Chen, K. Nagaraja, S. Li, and D. Raychaudhuri, “A Mobile Phone Based WSN Infrastructure for IoT over Future Internet Architecture,” *2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput.*, pp. 426–433, Aug. 2013.

[18] S. K. Datta, C. Bonnet, and N. Nikaein, “An IoT gateway centric architecture to provide novel M2M services,” *2014 IEEE World Forum Internet Things*, pp. 514–519, Mar. 2014.

[19] C.-H. Wang, “UPHSM: Ubiquitous personal health surveillance and management system via WSN agent on open source smartphone,” *2011 IEEE 13th Int. Conf. e-Health Networking, Appl. Serv.*, pp. 60–63, Jun. 2011.

[20] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, P. Dutta, and A. Arbor, “The Internet of Things Has a Gateway Problem.”

[21] P. Santos, L. Varandas, T. Alves, C. Romeiro, J. Casal, S. Lourenço, and J. Santos, “A Pervasive System Architecture for Smart Environments in Internet of Things Context,” in *ICMI 2015 : XIII International Conference on Multimodal Interaction, London, UK*, 2015.

[22] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang, “A Wearable Wireless Fall Detection System with Accelerators,” *2011 IEEE Int. Conf. Robot. Biomimetics*, no. 2, pp. 2259–2263, Dec. 2011.

[23] S. Hwang, M. Ryu, Y. Yang, and N. Lee, “Fall Detection with Three-Axis Accelerometer and Magnetometer in a Smartphone,” pp. 65–70.

[24] Q. T. Huynh, U. D. Nguyen, S. V Tran, A. Nabili, and B. Q. Tran, “Fall Detection System Using Combination Accelerometer and Gyroscope,” vol. 3200, pp. 978–981, 2013.

[25] A. J. Jara, L. Ladid, A. Skarmeta, I. Comsoc, and I. Etc, “The Internet of Everything through IPv6 : An Analysis of Challenges , Solutions and Opportunities,” *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 4, pp. 97–118.

[26] L. Tan, “Future internet: The Internet of Things,” *2010 3rd Int. Conf. Adv. Comput. Theory Eng.*, pp. V5–376–V5–380, Aug. 2010.

IEEE Systems Journal

Intelligent Personal Assistants Based on Internet of Things Approaches

João Santos, Joel J. P. C. Rodrigues, João Casal, Kashif Saleem

***Abstract*— The Internet has emerged as a key network to make information accessible quickly and easily, revolutionizing how people communicate and interact with the world. The information available on the Internet about a given subject may be extensive, allowing the development of new solutions to solve people's day-to-day problems. One such solution is the proposal of intelligent personal assistants (IPAs), which are software agents that can assist people in many of their daily activities. IPAs are capable of accessing information from databases to guide people through different tasks, deploying a learning mechanism to acquire new information on user performance. Repetitive interaction with the same user improves the IPA's behavior, resulting in a better understanding of the user's routines. IPAs can improve the assistance they offer to users by collecting information autonomously from objects that are available in the surrounding environment. To make this idea feasible, IPAs could be integrated into ubiquitous computing environments in an Internet of Things (IoT) context. Therefore, it is necessary to integrate wireless sensor networks (WSNs) with the Internet properly, considering many different factors, such as the heterogeneity of objects and the diversity of communication protocols and enabling technologies. This approach fulfills the IoT vision. This paper surveys the current state of the art of IoT protocols, IPAs in general, and IPAs based on IoTs.**

***Index Terms*— Internet of Things, Intelligent Personal Assistants, Artificial Intelligence, Wireless Sensor Networks, Ubiquitous Computing**

I. Introduction

THE importance of technology in human life has increased rapidly in recent decades. Technology now allows people to conduct their daily life activities in a simpler way, easing many tasks that once were very difficult to perform. One of the most important technological achievements was the creation of the Internet. It all began in the 1960s, when Leonard Kleinrock published the first paper on packet-switching theory [1]. His queuing theory demonstrated the theoretical efficiency of packet-switching communication, serving as a basis for the creation of the ARPANET – the Advanced Research Projects Agency Network [2], initially used only for military and research purposes.

The evolution of the Internet into the global communications network that it has become occurred over the years with the development of new paradigms and technologies, such as the Transmission Control Protocol/Internet Protocol (TCP/IP) architecture. TCP/IP, the computer networking model used in the Internet, was officially standardized in 1983. The importance of the Internet to human society in general began to be fully noticed in the 1990s with the creation and commercialization of the World Wide Web (WWW), which has revolutionized communications and social networking, allowing new forms of business, such as e-commerce; new forms of communication, such as e-mail or voice over Internet Protocol (VoIP) calls; and new forms of accessing information, such as through blogs, newspaper sites, and Internet search engines including Google. The Internet has massively increased the amount of accessible information and allowed the creation of new services to ease activities of daily life.

The evolution that has occurred in areas such as speech recognition, natural language processing, the semantic Web, machine learning, and artificial intelligence, combined with the huge amount of available information, has enabled the creation of intelligent personal assistants (IPAs), mobile, autonomous, software agents that are capable of performing tasks or services on behalf of humans [3]. An IPA is a multitasking machine where a single user command may trigger multiple processes. Using its learning mechanism and knowing its context, an IPA can respond autonomously to a user request, enhancing its capability to learn from repetitive actions.

The learning mechanism of IPAs could be improved by the integration of information generated from the surrounding environment, such as the temperature, humidity, and luminosity, among others. These external indicators could be measured with sensors that would then transmit the gathered values to the IPA. The values can be used by the IPA to generate useful information to trigger services to assist people with their daily life activities. This process can be seen as natural interaction among people, the environment, and machines, creating a scenario of ubiquitous computing.

A ubiquitous computing scenario occurs in an environment

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment.

João Santos, *Instituto de Telecomunicações*, University of Beira Interior, Covilhã, Portugal (e-mail: joao.santos@it.ubi.pt).

Joel J. P. C. Rodrigues is a Professor at *Instituto de Telecomunicações*, University of Beira Interior, Covilhã, Portugal and with University ITMO, St. Petersburg, Russia (e-mail: joeljr@ieee.org).

João Casal, *TIMWE I &D, Parkurbis – Parque de Ciência e Tecnologia da Covilhã, Portugal* (e-mail: joao.casal@timwe.com).

Kashif Saleem is with Center of Excellence in Information Assurance, King Saud University, Riyadh, Kingdom of Saudi Arabia (e-mail: ksaleem@ksu.edu.sa).

where technology is so integrated with users that they are unaware of the existence of the technological functions that surround them [4]. The Internet of Things (IoT) is a relatively recent technological paradigm that uses this concept of ubiquitous computing. The vision of IoT can be seen as the evolution of the Internet, as almost every object can support an IP address and consequently be connected to the Internet. Even tiny devices with high constraints in terms of memory, power consumption, energy autonomy, and communication capabilities can be part of an IoT network [5]. However, device heterogeneity combined with the communication protocol used, security issues, and network scalability pose problems in disseminating the IoT concept around the world.

Integrating IPAs with IoT objects can be a promising solution to offer people the perfect personal assistant, with the ability to act, manage, and interact autonomously with the environment and suggest suitable solutions to problems that arise in daily life. The current paper aims to review the state of the art of IPAs based on IoT solutions. A deep analysis of available approaches will be performed, and open issues will be identified. The main contributions of the paper are as follows:

- Presenting an overview of IoT technology and a review of the available enabling technologies and protocols;
- Showing the main features of IPAs by providing some examples of approaches involving these features;
- Describing how IoT networks may improve IPAs functionality.

The rest of the paper is organized as follows. Section 2 presents an overview of the most relevant aspects related to IoT, emphasizing the most significant protocols and corresponding research efforts. A general overview of IPAs, describing their importance in society, addressing some key contributions from other subjects, such as artificial intelligence (AI), multi-agent systems (MAS), and speech technologies, is given in Section 3. Some examples of projects using IPAs are also presented. Section 4 discusses the application of IPAs in ubiquitous scenarios, emphasizing the use of cloud computing. An analysis of the most relevant solutions is given in Section 5, and some open issues are identified. Section 6 concludes the paper by summarizing the main findings.

II. Internet of Things

The Internet revolution brought a new paradigm where people and machines can communicate among themselves. This gives some autonomy to machines (with the emergence of the machine-to-machine communication paradigm [6]), as many objects are connected through the network. This IoT vision *"brings life"* to almost every device by assigning an identification and an IP address to almost every object that can generate, pass on, or receive some kind of information [7][8][9][10].

Sensors integrated into objects can gather data from the environment (such as temperature or humidity data) that can be processed later. A wireless sensor network (WSN) can include a great number of nodes (dozens, hundreds, or millions) [11]. The ability to connect a WSN to the Internet provides a great opportunity to deliver information in real time. There are two main approaches to providing IP addresses for sensors and connecting the network to the Internet: (i) a proxy-based and (ii) a sensor node IP stack-based method [9]. The proxy-based approach uses a sink node that serves as a proxy, providing a bridging mechanism between a WSN and the Internet. The sensor nodes use dedicated protocols that interact with the sink node, which also supports the TCP/IP stack, enabling Internet communication. The main disadvantage of this approach is the gateway functionality, as it involves a complex development process and can be identified as a point of failure. In contrast, the sensor node IP stack-based approach relies on IP as the routing protocol inside the WSN, which allows direct communication among the nodes inside the WSN and with the Internet. A gateway can also be used to connect the WSN to the Internet, filtering all the unwanted traffic.

Various technologies that can support communication among objects, such as radio frequency identification (RFID) [12][13][14][15], near-field communication (NFC) [15][16][17], quick response (QR) codes [15], and bar codes [15]. RFID has been widely adopted in IoT projects because it has many advantages over other described technologies, such as higher memory capacity and the fact that it can be read by radio waves when the reader is not in a line of sight with the tag [12]. Each object using RFID has a unique tag attached to it to make it a unique, identifiable object.

With the creation of IoT networks, the amount of data generated has increased dramatically because the number of devices generating traffic inside the network much larger than before. The data can be used to create useful information, enabling the creation of new types of services and applications. IoT can have a big impact on a business by automating some processes and improving the control of many environment variables, easing the decision-making process.

Various projects have been designed to deploy IoT. One example is projects that automate processes inside a house – the smart home. In [18], the authors demonstrated some applications that integrate Web services, cloud computing, and IoT. The work presents three different case studies of smart home applications.

Another great opportunity enabled by IoT technology is the smart city, an urban infrastructure that can interact in real time with the surrounding environment autonomously and insightfully. Using low-power sensors, wireless networks, and Web and mobile-based applications, smart cities aim to use public resources in a more capable way, offering better services to citizens while decreasing the operational costs of those services. In [19], the authors discussed the protocols and technologies that are needed to construct an IoT urban structure. They also presented a real example of a smart city environment implemented in Padova, Italy and named the "Padova Smart City," where some environmental indicators are measured and controlled.

IoT can have a great impact on healthcare. As presented in

[20], the use of IP-equipped biomedical sensors can ease patient monitoring. Using technologies designed for these constrained devices (biomedical sensors), the authors developed a ubiquitous healthcare system, with the data routed to an IPv6 gateway connected to the Internet. Then, the data can be easily accessed by physicians on their PDAs. A solution to monitor continuously hospitalized patients using a wireless sensor network with mobility support was proposed in [21].

Another business area that can benefit from the IoT technology is manufacturing. In [22], the authors described a remote manufacturing monitoring system that uses sensors, RFID technology, and ZigBee, Wi-Fi, or Ethernet networks to gather information about industrial machines. The system is designed to collect data from industrial equipment (using sensors to detect temperature, humidity, vibration, pressure, etc.). Users can interact with the system through a Web browser to collect real-time data, alarms, and historical data queries, among other functions. The authors of [23] took a similar approach to implement a system to monitor a discrete manufacturing process. In this case, RFID was used to gather real-time information about system processes for the purpose of management and control.

Despite all the recent projects using IoT technology, a standard approach is not widely followed. An initial conceptual architecture has been accepted by the scientific community based on simple context factors. This architecture is composed of three layers, the perception layer, network layer, and application layer, as depicted in Figure 1. The work presented in [13] and [24] highlighted this three-layer IoT architecture as the initial architecture for the development of IoT projects.

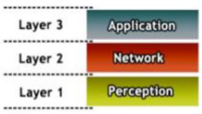

Fig. 1. Illustration of IoT Conceptual Architecture.

The perception layer considers the identification, acquirement, and collection of information by sensing the physical properties of objects using sensors, RFID tags, 1D and 2D bar codes, actuators, and other devices. The main purpose of the network layer is to transmit and process information. The convergence between IoT and industrial technologies occurs in the application layer, enabling real-time applications for different types of industries.

In this three-layer IoT architecture, security, reliability, and information processing are not considered. This led to the development of some other, more robust proprietary architectures. Examples include the ISA-100 [25], WirelessHART [26], ZigBee Pro [27][28], ZigBee IP [29][30], and Jennic IP [31] architectures. Some of these architectures use the IP protocol (ZigBee IP, for example), while others use proprietary protocols, such as WirelessHART.

To turn IoT into a feasible technology, many standardization of efforts have been performed. A standard for the physical and medium access control layers, the IEEE 802.15.4, has been approved. In addition, an overlay protocol was created to enable the fragmentation of the IPv6 packets inside IEEE 802.15.4 frames, called IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN). A routing protocol was defined for low-power and lossy networks (LLNs), referred to as the ripple routing protocol (RPL). A Web protocol was also created for LLNs, called the Constrained Application Protocol (CoAP). Besides CoAP, other protocols are being considered to serve as the standard IoT application layer protocol, such as the Extensible Messaging and Presence Protocol (XMPP) and the message queue telemetry protocol (MQTT).

A. IoT Protocols

Using power-constrained objects (such as sensors) on IoT networks can cause various performance issues. To attenuate these issues, the Institute of Electrical and Electronics Engineering (IEEE) created in 2003 the **standard IEEE 802.15.4**, which defines the physical (PHY) and medium access control (MAC) layers of low-rate wireless personal area networks (LR-WPANs) [10][32]. It has a maximum data rate of 250 kbps, much lower than the rate defined in the IEEE 802.11 standard. This fact allows small and power-constrained nodes to consume less energy. The standard IEEE 802.15.4 has a frame size up to 127 bytes, with two transmission modes, beacon-enabled mode and non-beacon-enabled mode, in the MAC layer. The CSMA-CD is used for collision avoidance in both cases, but in beacon-enabled mode, the device can go from idle to a low-power sleep state to save energy.

Devices using the IEEE 802.15.4 can communicate over areas covering 100 meters on single-hop architectures Transmissions beyond this range require the cooperation of neighbor nodes to deliver the message to its final destination using a multi-hop architecture. Two types of nodes can exist on a multi-hop architecture: a full-function device (FFD) and a reduced-function device (RFD). An FFD can served as the coordinator of a personal area network (PAN) and communicate with other nodes in the network. RFD has a lower processing capability, so it can never act as the PAN coordinator, communicating only with FFD nodes. There are two network topologies, the star topology and the peer-to-peer topology, as depicted in Figure 2.

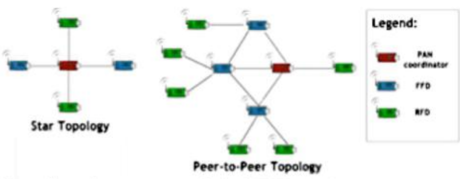

Fig. 2. Star and peer-to-peer topologies in IEEE 802.15.4.

IPv6 over Low Power Wireless Personal Area Networks

(6LowPAN), developed by the Internet Engineering Task Force (IETF), is an essential protocol for all IoT IP-based architectures using the IEEE 802.15.4 standard [32] [33].

Using IPv6 as the Internet protocol in IP-based IoT architectures allows the presence of more devices on thenetwork (2^{128}) compared to IPv4 (2^{32}). It also uses the stateless address auto configuration (SLAAC) mechanism to assign an IPv6 address to a device automatically at the moment of its initialization. Using the SLAAC mechanism on WSNs is very advantageous because, on these types of networks, some nodes can be physically inaccessible, making their manual configuration impossible [33]. However, Low Power Wireless Personal Area Networks (LoWPAN) have other characteristics that cannot be fulfilled by the use of IPv6. One example is the impossibility of directly encapsulating IPv6 packets (1,280 bytes MTU) in IEEE 802.15.4 frames. To solve this problem, the 6LoWPAN protocol was created; it compresses the size of packets flowing in the network, decreasing both the bandwidth and the energy consumed by power-constrained devices. Beyond adapting the data packets' size, 6LoWPAN is capable of resolving addresses, implement addressing management mechanisms, and discover devices and services.

Various open-source operating systems are available that implement the 6LoWPAN stack, such as the Contiki OS [34] with its uIPv6 stack or the TinyOS [35] with its BLIP implementation. Both implementations are important to the deployment of IoT projects using the 6LoWPAN protocol. For example, [36] demonstrated an application that enables an IPv6 connection between IPv6 and 6LoWPAN nodes using the Tiny OS 2.1 BLIP. A gateway is used to connect the 6LoWPAN nodes to the Internet. A 6LoWPAN implementation over low-power and economic CC430-based wireless sensor nodes is presented in [37], giving insight into the 6LoWPAN protocol and its importance to power-constrained devices on IoT deployments. The work also demonstrated some 6LoWPAN solutions that are currently available in the market.

In 2008, the Internet Engineering Task Force Working Group on Routing Over Low-Power and Lossy Networks (ROLL) was created. The ROLL Working Group intended to create a routing protocol for LLNs that would support a wide variety of link layers with common requirements such as low bandwidth, lossiness, and low power. The protocol was named the **ripple routing protocol (RPL)** [38][39]. The RPL is an IPv6 distance vector routing protocol, oriented for directed acyclic graph (DAG) topologies, in which no direct cycles connect the network devices. Generally, such networks have a root node or a set of root nodes to coordinate tasks and collect data. For each root node, a destination oriented directed acyclic graph (DODAG) is created using an objective function that defines how the routing metric is computed and *ranks* to encode the distance of each network node to its reference root. Figure 3 shows a DODAG topology with a root node and a set of neighbor nodes. The nodes in the topology have calculated their own distance to the root node of the tree, creating their own *ranks*.

Various types of messages are used to construct the network graph, called DIO (DODAG Information Object), DIS (DODAG Information Solicitation), and DAO (DODAG Destination Advertisement Object). DIO messages allow a node to discover an RPL graph, learn its configuration parameters, and select DODAG parents. A DAO message is used to spread destination information upward along the DODAG. A DIS message corresponds to the solicitation of a DIO from a RPL node. The RPL also uses mechanisms to repair the network graph when a failure on a link occurs, as well as an adaptive timer mechanism (the trickle timer) to control the rate at which DIO messages are sent to the network [40].

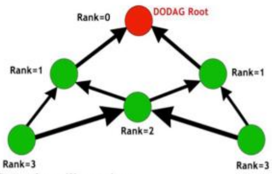

Fig. 3. DODAG topology illustration.

The RPL can be implemented in various operating systems, such as the Contiki OS and the TinyOS. The authors of [41] addressed the interplay between RPL and various address auto-configuration algorithms, using ContikiRPL to conduct the practical experiments. The work described in [42] evaluated the performance of 6LoWPAN and RPL protocols in TinyOS. The paper gives insight into the characteristics of the RPL standard and its relation with the TinyOS, comparing the TinyRPL solution with the de-facto standard routing protocol for TinyOS 2.X, the collection tree protocol (CTP).

IoT devices can be integrated on the Web, creating the "Internet of Things" concept. This allows Representational State Transfer (REST) architectures to be used on IoT applications [5][38][43]. Many of those REST architectures use HTTP, which may create performance issues when used on LLNs because of its large overhead. To overcome these issues, the **constrained application protocol (CoAP)** was created in 2010.

CoAP is based on the REST architecture using the PUT, GET, DELETE, and POST methods, which are also available on HTTP. Despite using the same methods, CoAP is not a substitute for HTTP, being easily interoperable through proxy mechanisms. As shown in Figure 4, the main difference is the use of UDP as the transport layer protocol on CoAP and the TCP on HTTP.

The application layer on CoAP is divided into two sub-layers: the message and request/response layers. The message layer controls message exchanges over UDP between two endpoints. Messages are specified by IDs to detect duplicates, assuming one of four different types: confirmable, non-confirmable, acknowledgment, and reset. The request/response layer is responsible for the transmission of requests and responses used to manipulate resources. A REST request is piggybacked on a confirmable or non-confirmable

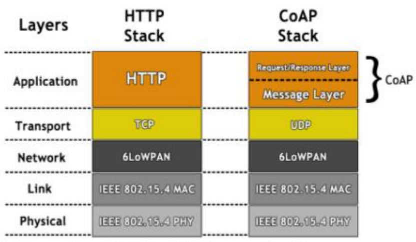

Fig. 4. Comparison of CoAP and HTTP protocol stacks.

message, while a REST response is related to an acknowledgment message [38]. When confirmable messages are used, acknowledgment messages are required to confirm the receipt of the desired information. If a response does not arrive at the client (or the client receives a reset message), the message is retransmitted. With this mechanism, CoAP ensures communication reliability without using TCP.

According to the literature, various CoAP deployments on IoT projects have been implemented. In [44], a comparison is made between CoAP and HTTP through the execution of experiments on client-server scenarios using CoAP in one scenario and HTTP in another scenario. It was concluded that CoAP has a better response time for client requests than HTTP. In [45], the authors presented an IP-based solution to integrate sensor networks properly in a cargo container with logistic processes, using CoAP to obtain data gathered by sensors (such as temperature or humidity data) during land or sea transportation. The authors of [46] proposed an IoT solution for the mining industry based on rock bolts and CoAP-enabled sensors. This proposal enables online and real-time monitoring of rock conditions inside mines, detecting potential anomalies at an early stage and raising security levels.

Standardized by the IETF, the **Extensible Message and Presence Protocol (XMPP)**, formerly known as Jabber, is based on Extensible Markup Language (XML), which allows two or more entities to communicate inside a network. XMPP is particularly suitable for real-time communication such as instant messaging. The XML data elements, known as "XML stanzas," enable communication inside the network, where, in general, a distributed client-server architecture is implemented.

XMPP is a scalable protocol that allows the specification of extension protocols, designated by XMPP extension protocols (XEPs). XEPs allow the addition of new functionalities to the XMPP protocol, introducing the ability to adapt to specific environments. Some XEPs (listed in [47]) can be useful for the IoT domain [48], including XEP-0323, used to provide the architecture, operations, and data structures for communication between sensors on XMPP networks; XEP-0324, used to manage access privileges and to ensure data delivery on existing services; XEP-0325, used to specify the control mechanism on IoT objects; and XEP-0326, which defines how to control architectures that contain servers or concentrators that handle multiple sensors.

The implementation of the XMPP in a IoT scenario is discussed in [49]. The authors presented a lightweight implementation of the XMPP for the Contiki OS, the µXMPP. The µXMPP is configured to follow LLN specifications, using only essential XMPP characteristics to reduce memory usage on the devices. The authors of [50] showed an improvement to the µXMPP solution with the addition of new modules to the µXMPP architecture, enabling IPv6 support, short JIDs, temporary subscription for presence (TSP) to reduce network traffic, and support for new XEPs, such as XEP-0045 for multi-user chat and XEP-0174 to allow communication without interference from the server. The work described [51] deployed a service-oriented solution for IoT scenarios, using XMPP as the communication protocol between entities in the network. It considers the diversity of network communication scenarios that exist in the real world and the services that are required by them, such as backend systems, mobile devices, and objects in the environment.

Message Queue Telemetry Protocol (MQTT), developed by the International Business Machines (IBM), is a lightweight application layer protocol based on a publish/subscribe method. With MQTT, clients can subscribe to information (data types) of interest by registering with a broker using TCP connections. Each message data type is referred to as a topic, and clients can subscribe to multiple topics. Every message sent on a topic is received by the clients who have subscribed to it. Figure 5 depicts communication between MQTT publishers and MQTT subscribers using a broker as an intermediary for the communication.

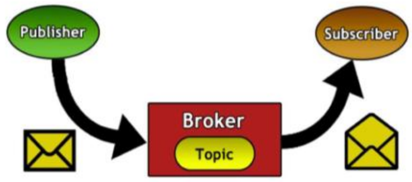

Fig. 5. Illustration of the MQTT publish/subscribe scheme.

The MQTT protocol is widely used in M2M communication, and it can be used in IoT environments. However, MQTT uses TCP as the transport layer protocol, provoking some inefficiency when used on power-constrained devices. This happens because of the extra power that these devices must consume when data must be retransmitted. To better adapt the MQTT protocol to LoWPANs, MQTT-S (MQTT for sensor networks) was created; it defines a UDP mapping for the MQTT, adding broker support for indexing topic names at the same time. The MQTT-S network architecture is constituted by four different elements: a MQTT-S client, a MQTT broker, a MQTT-S gateway, and a MQTT-S forwarder. A MQTT-S client connects with the MQTT broker using the MQTT-S protocol. It is used a MQTT-S gateway, which can be incorporated into the broker itself or implemented in the LoWPAN edge router. The gateway makes the correct translation between TCP and UDP,

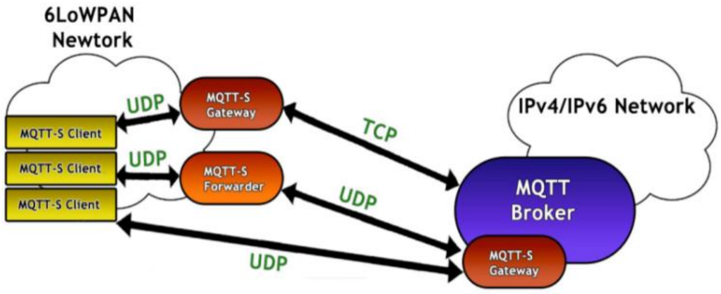

Fig. 6. Illustration of the MQTT-S publish/subscribe scheme.

allowing connections between devices on an IPv6 network and an IPv4 network. If the gateway is not directly accessible, a MQTT-S forwarder is used to send messages from the clients to the broker. The MQTT-S architecture can be described as shown in Figure 6 [52].

MQTT usage in IoT environments was shown in [53], which examined the implementation of MQTT-S in LLNs. The authors discussed the importance of the publish/subscribe method for WSNs and applications where the content itself is more important than its origin. One of the highlighted MQTT-S characteristics is the possibility of using multiple gateways through a gateway discovery process, which is desirable when link failure scenarios occur. To test the MQTT-S architecture, consisting of a MQTT-S client, a message broker, and a MQTT-S gateway, the authors used the IBM wireless sensor networking testbed [54] with two types of clients: ZigBee clients and TinyOS clients.

III. Intelligent Personal Assistants

Personal assistants are not recent, having been used in daily business and personal tasks for many years. Personal assistants are responsible for daily task management such as scheduling meetings, reserving hotel rooms, shopping, and paying bills, among other tasks. In the past, the role of a personal assistant was performed exclusively by people; nowadays, the personal assistant role can be performed by a digital device with learning capabilities.

In the 1990s, the IPA concept was introduced. As stated in [55], the use of techniques from the **artificial intelligence (AI)** field created the possibility of constructing intelligent machines that are capable of autonomously performing tasks on a user's behalf. The use of mechanisms such as data mining and machine learning algorithms [56] was important in creating more responsive and self-aware machines. Machine learning algorithms can be divided into two different categories: supervised learning [57] and unsupervised learning [58] methods. Supervised learning assumes the existence of a labeled training set composed of input-output pairs that make possible the prediction of a new output value from a new input. Supervised learning models include learning decision trees, Bayesian classifiers, linear regression, case-based reasoning (CBR), and neural networks. The unsupervised learning method is more complex, as the training data is not labeled and the machine receives only the inputs. In this model, machines must build representations of the inputs to help in the prediction of new incoming values. The process is based on a search for patterns that can be exploited to create knowledge from input values. Some examples of unsupervised learning models are the K-means clustering algorithm, hidden Markov models (HMMs), and *a priori* algorithms.

IPAs can normally interact equally with other intelligent objects in the environment, human or machine, to obtain knowledge about different domains. These scenarios are referred to as **multi-agent systems (MASs)** [59][60]. MASs are composed of multiple heterogeneous interactive intelligent agents within an environment, enabling parallel processing inside the system and making it less prone to failures. In a MAS, agents in the environment can be influenced by other agents or by the environment itself. In single-agent systems (SASs), only the environment can influence the behavior of intelligent agents in the environment. In a SAS, an agent sees another agent as an integral part of the environment, ignoring the fact that other agents are individual entities in reality. Figure 7 illustrates a scheme that represents the difference between MAS and SAS scenarios.

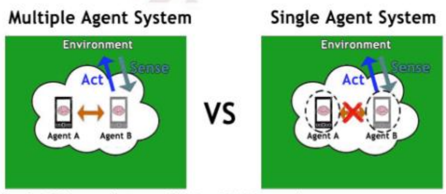

Fig. 7. Difference between MAS and SAS scenarios.

The way in which intelligent agents in an environment relate among themselves is important in developing complex systems. First, it is necessary to identify the contribution that each individual agent can make to the overall intelligence of the system. This process can be conducted by isolated learning. The learning that agents in the environment perform as a group can be referred to as collective learning. Combining the two types of learning mechanisms and MAS makes possible the design of complex, autonomous, and self-aware computational systems. Intelligent agents in MAS scenarios can gain awareness of what they must do under the influence of other agents in the environment, the environment itself, and their own prior knowledge about multiple subjects. It is like having a network of self-conscious machines that can make their own decisions in a way that seems natural by human standards. This fact may be useful in the construction of optimized IPAs.

The technological revolution has enabled the creation of new gadgets with higher computing capabilities and new features, such as smartphones. Some smartphones provide a personal digital assistant as one of their main features. Apple's Siri [61], Google Now [62], Samsung's S Voice [63], LG's Voice Mate [64], and Microsoft's Cortana [65] are examples. Generally, these digital assistants make use of **natural language user interfaces (NLUIs)** to interact with users. NLUIs allow human-machine interaction through the translation of voice commands into machine-level commands, an important **speech technology** for human-to-machine (H2M) communication.

The act of making a machine understand human language poses many issues, most of them related to the AI field. In 1952, Bell Laboratories designed a system, known as "Audrey," that recognized digits spoken by a single voice [66]. This experiment focused only on the understanding of numbers since human language understanding was little explored at that time. Other speech recognition experiments were conducted in the 1950s and 1960s, such as IBM's 1962 effort. This system, called "Shoebox," could recognize and respond to 16 words spoken in English.

The 1970s marked the founding of the first commercial speech recognition company, the Threshold Technology, which commercialized the first real automatic speech recognition system, the VIP-100 System. In the same decade, the United States Department of Defense developed a speech recognition system called "Harpy" that could understand up to 1,011 words [66].

The 1980s brought great developments in the speech recognition field. A new statistical method to recognize words, hidden Markov models (HMMs), began to replace the templates used to associate words with sound patterns. HMMs are based on the probability of unknown sounds being words, extending the speech recognition vocabulary. However, human-machine dialogue was not optimal since speech recognition systems required discrete dictation.

The subsequent decades brought great development in the computational processing field with the emergence of computers equipped with faster processors and higher memory capacity. This led to the construction of bigger dictionaries to accommodate human vocabulary known by machines, increasing their knowledge. The processing of words by machines was improved, as they became able to understand continuous speech rather than just isolated words. The development in areas such as automatic speech recognition, natural language understanding, and text-to-speech synthesis were very helpful in constructing today's applications.

The creation of new algorithms to increase the context awareness of machines, support for multiple languages, and the development of new mechanisms to recognize specific voices are future steps to improve speech recognition technology. Figure 8 shows a communication scenario between a user and the Samsung personal assistant, S Voice.

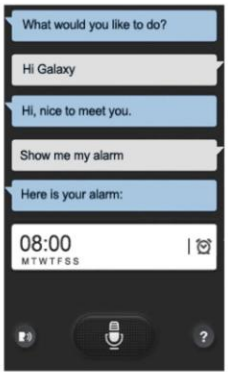

Fig. 8. Interaction between a user and the S Voice assistant.

A. Case Studies

H2M communication is a field of great interest in the scientific community, with projects involving IPAs and similar devices existing in great numbers. The authors of [67] discussed existing issues in designing a personal assistant, illustrating some mathematical models to resolve these issues. The authors in [68] presented an intelligent assistant system composed of intelligent software agents that can help the user with communication, information, and time management. Each intelligent agent is responsible for a different task, such as calendar scheduling and telephone call filtering, among others. An intelligent assistant for athletes was proposed in [69], called iAPERAS. It aims to help non-professional athletes who cannot afford their own training team to obtain dedicated information about health indicators. To improve the quality of the information assessment, the iAPERAS system uses Bayesian networks, relying on scientific research findings. The authors of [70] proposed an IPA that allows doctors to monitor the elderly. The IPA, called HealthPal, helps elderly people monitor their health status autonomously, interacting with devices such as thermometers, blood pressure measuring systems, and PDA applications. The medical records are stored on the user's PDA and on the doctor's PC. Notifications are generated when a health problem is detected in the patient. The authors of [71] presented a framework for an intelligent assistant system based on CBR and MAS for manufacturing. The MAS framework allows the creation of a collaborative environment between intelligent agents, decentralizing the processing in the system. Thus, the subsystems throughout the manufacturing system can be seen as individual, intelligent, and collaborative agents. The CBR technique is used to find patterns in the information that is presented based on past experiences. The use of CBR allows the system to find previous orders that are similar to those placed at the time. The authors of [72] presented a pedagogical agent designed to work with Web-based instructional material, called “Adele.” This intelligent agent aims to help students learn new subjects by presenting them with some practical cases to solve. The Adele agent is also supposed to collaborate with students by making hints, posing questions, and providing explanations. The authors of [73] presented an intelligent assistant for network based communications, discussing the importance of creating a context-aware assistant that can manage communication within the telecommunications network. The proposed agent is based on the Session Initiation Protocol (SIP), which can interact with external information and application servers. The authors of [74] presented a speech interface for personal assistants used in research and development (R&D) projects. The speech interface presented in the paper is aimed for use in cooperative scenarios with multiple agents. It is used in a knowledge management (KM) multi agent system in which the user can talk in English to the personal assistant to perform various tasks, such as listing all articles available on a certain subject in a database. The personal assistant is supposed to understand what the user says through the use of an ontology handling mechanism that employs the Protégé [75] tool. This paper shows that it is possible to construct a speech recognition interface for IPAs based on projects of different scopes. The authors of [76] proposed a personal assistant to execute services on behalf of users, using natural language interfaces to process the users' requests. For each request, the personal assistant analyzes the phrase and segments it into individual words. Then, it tries to find keywords in the phrase, and a suitable service for the most meaningful keyword is executed. The authors of [77] proposed an IPA that can learn the preferences, goals, and habits of the user by using supervised learning mechanisms, based on the user's previous experiences. The process of monitoring the users is done using Aspect Oriented Programming, breaking down the program logic. Traces are generated on the user's actions to ascertain whether he/she has completed the desired task. If the task is completed, the traces are successful; otherwise, the traces are unsuccessful. To enable the IPA to predict the user's actions, the authors used a feedforward neural network, trained with a backpropagation algorithm.

IV. Intelligent Personal Assistants and the Internet of Things

Currently, IPAs are already capable of helping their users perform several tasks. It is necessary to ascertain whether it is possible to improve their knowledge and autonomy by using new technological paradigms. One possible solution would be the insertion of IPAs in ubiquitous communication scenarios. It would enable IPAs to analyze information from a greater variety of sources, increasing the learning database. Such scenarios are possible with the usage of IoT technology.

With the creation of IPAs that are more autonomous and insightful, many tasks performed by users can be eased. Take, for example a smart home scenario, where an IPA can communicate with other objects in the environment: a person wakes up at 7:00 am and must prepare to go to work. As the alarm clock rings, the curtains in the user's bedroom slowly open to allow some light to enter the room. Three minutes after the alarm clock stops, the person goes to the bathroom to take a shower. After the person takes his shower, the toaster and the coffee machine in the kitchen turn on automatically to prepare breakfast. All these actions are transparent to the user, as it is the IPA that autonomously prepares the user's morning routine. This is possible because the IPA learns the actions that the user normally performs in his daily routine, either by checking the interaction of the user with the IPA itself or through the interaction between the user and other objects in the environment. It is in the cooperation between IPAs and smart objects that the IoT concept is important. Figure 9 depicts the interaction among the IPA, the user, and the smart objects inside a house.

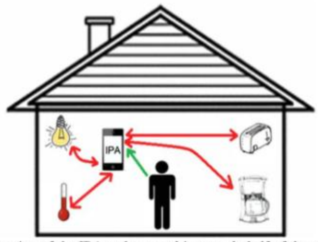

Fig. 9. Interaction of the IPA and smart objects on behalf of the user.

However, some issues arise from the idea of introducing IPAs in IoT environments, mainly related to technological paradigms. One of these issues is related to the creation of mechanisms to support communication between objects within the network. One possible solution is the creation of a standard IoT communication protocol. Until then, other solutions can be attempted, such as the creation of **network gateways** to map different protocols and networks. The interconnection between IoT environments and external networks has been a subject of research in recent years. The authors of [78] proposed a gateway prototype to connect WSNs with the Internet. The WSN is composed of 6LoWPAN nodes that use short addresses to communicate. The gateway can map these 6LoWPAN addresses into full IPv6 addresses, establishing the connection between the WSN and the Internet. The authors of [79] described the procedures necessary to design a gateway capable of connecting a 6LoWPAN network to an IPv6 network. The solution included the use of low-power sensor nodes configured with the TinyOS 6LoWPAN protocol stack and an IPv6 client to interact with the same sensor nodes. The gateway was important in compressing the IPv6 header when the communication was done from the IPv6 client to a WSN node and in decompressing the 6LoWPAN header when the communication flowed from the 6LoWPAN WSN. The authors of [80] discussed a gateway that connects 6LoWPAN networks to IPv4/IPv6 networks. The gateway was used to map 6LoWPAN addresses to IPv4\IPv6 addresses and vice versa, building a ubiquitous communication scenario. The authors of [81] described the construction of a prototype for a gateway that connects IEEE 802.15.4 nodes with Ethernet IPv6 networks. The performance of the gateway was evaluated through its capability to compress and decompress IPv6 packets.

With more objects communicating in the same environment, more data are generated. Some of that data may be important to the application scope, making it necessary to store and to manage them. The necessary storage space will be enormous, so new mechanisms must be applied. The use of **cloud computing** technology can be very useful in this case, decentralizing the processing involved in storing and managing the data. The authors of [82] discussed reasons to integrate IoT in the cloud and the difficulties that arise from that integration. The integration of IoT in the cloud led to the development of the concept of CloudIoT, the creation of mechanisms to store data and share computational resources, and a medium where heterogeneous objects can communicate as well. Possible challenges arise from the lack of solutions to manage all the data generated in the IoT network inside the cloud and the lack of a service to catalog the data generated by all the heterogeneous IoT devices in different categories. The authors of [83] presented a cloud-based IoT architecture, describing a smart home environment containing PCs, PDAs, indoor and outdoor sensors, microwave ovens, toasters, televisions, and stereos, among other appliances. The interaction between those objects leads to some automatic actions that interfere with the daily life of the people who live in the house. The cloud infrastructure is used to store data coming from the objects. The authors of [84] addressed the feasibility of the integration of an IoT network with the Internet using various cloud services. They also described various technologies that can enable the construction of IoT networks, such as WSNs, next-generation networks, and cloud computing platforms. Furthermore, they tested an IoT cloud platform named Skynet, an instant messaging platform oriented for M2M communication based on the MQTT protocol. The authors of [85] described how IoT and the cloud computing technologies can work together to deal with the huge amount of data that exists in the network. A solution where the sensing of a certain physical measure is considered a service on the cloud was proposed. This allows the creation of a ubiquitous communication scenario where the sensors are the interface between the physical world and the digital world.

In a scenario where an IPA can communicate with every object that surrounds it, extra care must be applied to prevent private information from falling into the wrong hands. These **security** issues are discussed in [86]. The authors identified five IoT security issues related to (i) the correct identification of IoT objects, (ii) the wireless link insecurities originated by each object connected to the IoT network, (iii) the heterogeneity of network devices and their specific security needs, (iv) the confidentiality of the information flowing on the IoT network, and (v) the information processing security. All these issues can be applicable to scenarios where IPAs are inserted into ubiquitous communication environments, as IPAs can behave as smart objects in an IoT environment. The authors of [87] identified the main problems that can emerge from the creation of IoT networks. Because IoT networks cover a wide range of devices and a great amount of traffic, the authors assumed that it is not possible to use the security mechanisms currently used on the Internet in IoT networks. Problems such as protocol and network security, as well as privacy and data and identity management, were enumerated. An IoT scenario where sensors integrated into a smart city environment must send data to a mobile device (like a PDA) was given as an example. In this scenario, new cryptography algorithms, key management systems, and security protocols would be needed to secure the communication. The authors of [88] provided insight into the efforts made by the IETF to develop a standard solution to secure the communication inside an IoT network. The solution uses the CoAP protocol along with Datagram Transport Layer Security (DTLS). Because it is not directly related to existing IPAs , this paper is important in understanding the procedures that must be implemented to introduce IPAs on a secure IPv6 network.

IPAs can be used in many fields for various purposes. For example, The authors of [89] described a solution that uses a wireless body sensor network (WBSN) to monitor the vital signs of multiple patients. The information is passed to the intelligent personal digital assistants (IPDAs) of the medical staff using the ZigBee/IEEE 802.15.14 standard. The IPDA must analyze the data to correctly determine the health status of the patients. The authors of [90] presented a WSN for health monitoring. The system includes wearable devices, with some of them placed on the body of the patient and others placed around the house. The medical staff is informed about the health status of the patients through notifications received on their PCs and PDAs. The sensors that constitute the network architecture are connected to the backbone (PCs, PDAs, and databases) through motes that use the IEEE 802.15.4 protocol.

Other types of applications that can be enhanced include smart home applications. The AlertMe [91] and Iris [92] systems are two solutions that can manage home appliances through smartphones. The authors of [93] proposed a remote monitoring and controlling system composed of two sub-systems. One of the sub-systems is used for the real-time monitoring of the house, while the other is used to control the lights in the house, implementing a ZigBee network. The user can see real-time video footage of the house or control the lights by accessing the Web application or the smartphone application. The authors of [94] described a smart home system composed of an application to control and monitor home appliances, developed for the Android operating system, and an Arduino Ethernet-based micro-Web server. All the sensors in the smart home environment connect with the Arduino platform. The system includes some security features, such as user authentication and the ability to notify the user in the case of any unexpected event (a fire in the kitchen, for example). The authors of [95] proposed a novel mechanism for a home appliance control system. The system's user can control the home appliances through a PDA, mobile phone, or notebook running the system control application. The home appliances are inserted into a 6LoWPAN WSN, where an appliance control terminal is used to manage each appliance. The system also includes a smart gateway to allow communication between the WSN and other networks.

V. Discussion and Open Issues

IoT will allow scenarios of ubiquitous communication by connecting every device in a common network. This means that devices that are supposed to interact directly with humans, such as IPAs, can also be included in IoT environments. Through interaction with other objects, IPAs can gain better knowledge about their users and consequently offer better assistance to them. However, an important question must be raised before deploying IoT scenarios: how will all the heterogeneous objects (including IPAs) communicate among themselves?

This is a big issue that currently has no definitive answer. The creation of a standard protocol to be used by all devices, including power-constrained ones, would enable the creation of ubiquitous communication scenarios with millions of devices. There are various candidates, such as CoAP [reference], XMPP [reference], and MQTT [reference]. These protocols are suitable for use in large networks based on small and low-power nodes, but they differ in some important respects. For example, in an IP-based protocol stack, there are two major protocols used in the transport layer: TCP and UDP. TCP has a larger overhead and consumes more energy but ensures that data are delivered to the destination. UDP presents a smaller overhead and consumes less energy but is not as reliable as TCP. **Power consumption** is one of the key aspects of IoT objects. With this in mind, among CoAP, XMPP, and MQTT, the most suitable protocol for the IoT would be CoAP, as it is the only one that uses UDP. MQTT has the MQTT-S [reference] version that uses UDP, but its architecture is more complex, as it requires specific gateways and brokers. Furthermore, UDP is a connectionless protocol, making it more suitable for larger networks.

Security is another factor that is very important for IoT networks (and others). In terms of security, TCP is better than UDP, as the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) are deployed on top of TCP. UDP also provides security mechanisms with the usage of the Datagram Transport Layer Security (DTLS) but it is heavier than TLS. XMPP and MQTT can provide better security mechanisms than CoAP for IoT networks. It is clear that a definite answer regarding the choice of IoT protocol cannot be easily obtained. The introduction of IPAs in IoT networks will create countless possibilities for services that can improve human life, but some issues need to be overcome. Some of the issues are related to IoT networks, while others are related to IPAs. Some open issues are as follows:

- Standardization of a communication protocol for IoT devices.
- Creation of mechanisms to store all the data generated in IoT networks.
- Improvement of the security mechanisms used in the network, considering that the growing number of communicating objects increases the potential sources of cyber attacks.
- Validation of devices that want to join the network. One of the conditions that IoT networks should respect is the low level of involvement of users in the configuration of devices. Therefore, users should perform very few configurations on their IPAs, meaning that IPAs should be self-configurable. This fact highlights the necessity to create a transparent mechanism that allows the user to validate and manage the smart objects in the network.
- Creation of better speech recognition mechanisms to address all the current problems in the recognition of voice commands made by users to their IPAs, such as the noise in the environment.
- Improvement of the context awareness on IPAs.

In a near future, with all these issues addressed correctly, IPAs can offer greater functionality. The opportunity to use IPAs in any desired context can be created, giving humans the opportunity to have a smart personal assistant to complete almost every task that they need to perform.

VI. Conclusions

This paper has reviewed the literature on IPAs in ubiquitous environments in the IoT context. IoT technology will enable the creation of ubiquitous communication scenarios, where almost all the devices in the environment will be able to communicate. A standard protocol for enabling communication in IoT networks would ease the creation of more ubiquitous communication solutions. Nevertheless, the available protocols, such as CoAP, XMPP, MQTT, and other proprietary protocols, allow the development of interesting IoT projects, such as smart city scenarios and smart home appliances.

IoT technology offers new opportunities, such as the creation of IPAs that can assist their users while communicating with other smart objects in the environment. This can increase the knowledge of IPAs, as they would learn the behavior of their users through direct interactions with them and through interactions with other smart objects in the environment. To achieve this goal, it is necessary to develop new mechanisms to increase the intelligence of IPAs, such as new machine learning algorithms and speech recognition mechanisms. The inclusion of new types of devices in the network, such as devices with embedded sensor units, will increase the heterogeneity of the network, increasing the generated traffic as well. To address these issues, it is necessary to take into account the creation of new security mechanisms to reduce cyber attacks and problems related to entity authentication. Another issue is related to the creation of mechanisms to manage all the data generated in the network.

The creation of a scenario where machines interact with humans in a smart and fully aware environment may not be as far away as it is thought. It is a good challenge, and the research community is interested in it.

ACKNOWLEDGEMENTS

This work has been partially supported by *TIMWE Investigação e Desenvolvimento*, through the AMBRO project, co-funded by COMPETE/QREN/EU, in the context of the Portuguese *Sistema de Incentivos à I&DT* (project 33779), by the Visiting Professor Program at King Saud University, by Government of Russian Federation, Grant 074-U01, by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project.

REFERENCES

[1] L. Kleinrock, "Information Flow in Large Communication Nets." .

[2] B. M. Leiner, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A Brief History of the Internet Professor of Computer Science," vol. 39, no. 5, pp. 22–31, 2009.

[3] P. Garrido, F. J. Martinez, and C. Guetl, "Adding Semantic Web Knowledge to Intelligent Personal Assistant Agents."

[4] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, 2001.

[5] A. J. Jara, L. Ladid, A. Skarmeta, I. Comsoc, and I. Etc, "The Internet of Everything through IPv6 : An Analysis of Challenges , Solutions and Opportunities," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 4, pp. 97–118.

[6] I. Stojmenovic, "Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 122–128, Apr. 2014.

[7] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," *2014 IEEE World Forum Internet Things*, pp. 287–292, Mar. 2014.

[8] "Cisco Homepage." [Online]. Available: http://www.cisco.com/web/solutions/trends/iot/overview.html.

[9] J. J. P. C. Rodrigues and P. A. C. S. Neves, "A survey on IP-based wireless sensor network solutions," *Int. J. Commun. Syst.*, vol. 23, no. January, pp. 963–981, 2010.

[10] L. M. L. Oliveira, A. F. De Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks," no. February, pp. 1445–1466, 2011.

[11] "EDN Networks Homepage." [Online]. Available: http://edn.com/design/wireless-networking/4428131/1/Designing-IoT-Part-1-IoT-Devices-and-Local-Networks.

[12] H. Zhang and L. Zhu, "Internet of Things: Key technology, architecture and challenging problems," *2011 IEEE Int. Conf. Comput. Sci. Autom. Eng.*, pp. 507–512, Jun. 2011.

[13] C. Xiang, "General analysis on architecture and key technologies about Internet of Things," *2012 IEEE Int. Conf. Comput. Sci. Autom. Eng.*, pp. 325–328, Jun. 2012.

[14] M. Zhang, F. Sun, and X. Cheng, "Architecture of Internet of Things and Its Key Technology Integration Based-On RFID," *2012 Fifth Int. Symp. Comput. Intell. Des.*, pp. 294–297, Oct. 2012.

[15] M. Vazquez-briseno, F. I. Hirata, J. D. D. Sanchez-lopez, E. Jimenez-garcia, C. Navarro-cota, and J. I. Nieto-hipolito, "Using RFID / NFC and QR-Code in Mobile Phones to Link the Physical and the Digital World."

[16] "Venturebeat Homepage." [Online]. Available: http://venturebeat.com/2011/06/21/nfc-and-the-internet-of-things/.

[17] S. Forsstrom, T. Kanter, and P. Osterberg, "Ubiquitous Secure Interactions with Intelligent Artifacts on the Internet-of-Things," *2012 IEEE 11th Int. Conf. Trust. Secur. Priv. Comput. Commun.*, pp. 1520–1524, Jun. 2012.

[18] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart Home: Integrating Internet of Things with Web Services and Cloud Computing," *2013 IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, pp. 317–320, Dec. 2013.

[19] A. Zanella, S. Member, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," vol. 1, no. 1, pp. 22–32, 2014.

[20] D. Singh and U. S. Tiwary, "IP-based ubiquitous healthcare system," *2008 Int. Conf. Control. Autom. Syst.*, pp. 131–136, Oct. 2008.

[21] A. June, "Healthcare Monitoring System Using Wireless Sensor Networks," *Int. J. Adv. Res. Comput. Sci. Technol.*, vol. 2, no. 2, pp. 443–446, 2014.

[22] Z. Xiang-li, Y. Jin, Y. Kun, and L. Jian, "A remote manufacturing monitoring system based on the Internet of Things," *Proc. 2012 2nd Int. Conf. Comput. Sci. Netw. Technol.*, pp. 221–224, Dec. 2012.

[23] L. Yuan, Y. Guo, J. Jiang, and L. Nian, "The Research on Monitoring of Discrete Manufacturing Process Based on Internet of Things," *2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput.*, pp. 1186–1191, Aug. 2013.

[24] M. Wu, T. Lu, F. Ling, and H. Du, "Research on the architecture of Internet of Things," *2010 3rd Int. Conf. Adv. Comput. Theory Eng.*, pp. V5–484–V5–487, Aug. 2010.

[25] "ISA 100 Homepage." [Online]. Available: http://www.isa100wci.org/.

[26] "WirelessHART Homepage." [Online]. Available: http://www.hartcomm.org/index.html.

[27] "ZigBee Alliance Homepage." [Online]. Available: http://www.zigbee.org/Products/CompliantPlatforms/ZigBeePROFeatureSet.aspx.

[28] "NXP Homepage." [Online]. Available: http://www.nxp.com/documents/user_manual/JN-UG-3048.pdf.

[29] "ZigBee Alliance Homepage." [Online]. Available: http://www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx.

[30] "Embedded.com Homepage." [Online]. Available: http://www.embedded.com/design/connectivity/4419558/1/Zigbee-s-new-IP-specification-for-IPv6-6LoPAN-wireless-network-designs.

[31] "Jennic Homepage." [Online]. Available: http://www.jennic.com/index.php.

[32] X. Ma and W. Luo, "The Analysis of 6LowPAN Technology," *2008 IEEE Pacific-Asia Work. Comput. Intell. Ind. Appl.*, no. 3, pp. 963–966, Dec. 2008.

[33] "IPSO Homepage." [Online]. Available: http://www.ipso-alliance.org/wp-content/media/6lowpan.pdf.

[34] "Contiki OS Homepage." [Online]. Available: http://www.contiki-os.org/.

[35] "Tiny OS Homepage." [Online]. Available: http://www.tinyos.net/.

[36] L. M. L. Oliveira, J. J. P. C. Rodrigues, B. M. Mação, P. A. Nicolau, L. R. Wang, and L. Shu, "End-to-End Connectivity IPv6 over Wireless Sensor Networks," 2011.

[37] B. Pediredla, K. I.-K. Wang, Z. Salcic, and A. Ivoghlian, "A 6LoWPAN implementation for memory constrained and power efficient wireless sensor nodes," *IECON 2013 - 39th Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 4432–4437, Nov. 2013.

[38] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, S. Member, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," vol. 15, no. 3, pp. 1389–1406, 2013.

[39] 39. B. A. Klein, "RPL : IPv6 Routing Protocol for Low Power and Lossy Networks," no. July, pp. 59–66, 2011.

[40] "cs.berkeley.edu." [Online]. Available: http://www.cs.berkeley.edu/~jwhui/6lowpan/IPSO-WP-7.pdf.

[41] E. Ancillotti, R. Bruno, and M. Conti, "On the interplay between RPL and address autoconfiguration protocols in LLNs," *2013 9th Int. Wirel. Commun. Mob. Comput. Conf.*, pp. 1275–1282, Jul. 2013.

[42] J. Ko and D. Culler, "Evaluating the Performance of RPL and 6LoWPAN in TinyOS," 2011.

[43] C. Bormann, U. Bremen, and Z. S. Sensinode, "CoAP : An Application Protocol for Billions of Tiny Internet Nodes," 2012.

[44] V. D. Walter Colitti, Kris Steenhaut, Niccolò de Caro, Bogdan Buta, "Evaluation of Constrained Application Protocol for Wireless Sensor Networks."

[45] K. Kuladinithi and T. Pötsch, "Implementation of CoAP and its Application in Transport Logistics."

[46] J. Eliasson, J. Delsing, A. Raayatinezhad, and R. Kyusakov, "A SOA-based framework for integration of intelligent rock bolts with internet of things," *2013 IEEE Int. Conf. Ind. Technol.*, pp. 1962–1967, Feb. 2013.

[47] "XMPP Standards Foundation Homepage." [Online]. Available: http://xmpp.org/xmpp-protocols/xmpp-extensions/.

[48] "XMPP Wiki." [Online]. Available: http://wiki.xmpp.org/web/Tech_pages/IoT_XepsExplained.

[49] A. Hornsby and E. Bail, "μXMPP: Lightweight implementation for low power operating system Contiki," *2009 Int. Conf. Ultra Mod. Telecommun. Work.*, pp. 1–5, Oct. 2009.

[50] R. Klauck and M. Kirsche, "Chatty Things - Making the Internet of Things Readily Usable for the Masses with XMPP," *Proc. 8th IEEE Int. Conf. Collab. Comput. Networking, Appl. Work.*, pp. 60–69, 2012.

[51] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, and M. Ameling, "A Service Infrastructure for the Internet of Things based on XMPP," no. March, pp. 385–388, 2013.

[52] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. .

[53] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," *2008 3rd Int. Conf. Commun. Syst. Softw. Middlew. Work. (COMSWARE '08)*, pp. 791–798, Jan. 2008.

[54] S. Furrer, W. Schott, H. L. Truong, and B. Weiss, "The IBM Wireless Sensor Networking Testbed," 2005.

[55] P. Maes, "Agents that reduce work and information overload," *Commun. ACM*, vol. 37, no. 7, pp. 30–40, Jul. 1994.

[56] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. 2010.

[57] S. B. Kotsiantis, "Supervised Machine Learning : A Review of Classification Techniques," vol. 31, pp. 249–268, 2007.

[58] Z. Ghahramani, "Unsupervised Learning," pp. 1–32, 2004.

[59] P. Stone, P. Ave, and F. Park, "Multiagent Systems : A Survey from a Machine Learning Perspective," pp. 1–57, 2000.

[60] G. Weib, "Learning to Coordinate Actions in Multi-Agent Systems," pp. 311–316, 1991.

[61] "Apple Siri Webpage." [Online]. Available: https://www.apple.com/ios/siri/.

[62] "Google Now Webpage." [Online]. Available: http://www.google.com/landing/now/.

[63] "Samsung S Voice Webpage." [Online]. Available: http://www.samsung.com/global/galaxys3/svoice.html.

[64] "LG Voice Mate Webpage." [Online]. Available: http://www.lg.com/us/support/product-help/doc-1403913325190-en-LGEAI.

[65] "Microsoft Cortana Webpage." [Online]. Available: http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana.

[66] B. H. Juang and L. R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology Development," pp. 1–24.

[67] C. Pozna, P. Foldesi, and J. Kovacs, "The Personal Assistant application, problem definition," *2013 IEEE 4th Int. Conf. Cogn. Infocommunications*, pp. 851–856, Dec. 2013.

[68] B. Azvine, D. Djian, K. C. Tsui, and W. Wobcke, "The Intelligent Assistant: An Overview."

[69] M. Verlic and M. Mertik, "iAPERAS – Intelligent Athlete ' s Personal Assistant," in *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, 2005, vol. 386, no. 0.

[70] A. Komninos and S. Stamou, "HealthPal : An Intelligent Personal Medical Assistant for Supporting the Self-Monitoring of Healthcare in the Ageing Society."

[71] F. Li and N. Chen, "An intelligent assistant design system based on CBR and MAS," *2009 IEEE 10th Int. Conf. Comput. Ind. Des. Concept. Des.*, pp. 1335–1339, Nov. 2009.

[72] E. Shaw, W. L. Johnson, and R. Ganeshan, "Pedagogical Agents on the Web," *Proc. third Annu. Conf. Auton. Agents - AGENTS '99*, pp. 283–290, 1999.

[73] B. El Saghir and N. Crespi, "An Intelligent Assistant for Context-Aware Adaptation of Personal Communications," *2007 IEEE Wirel. Commun. Netw. Conf.*, pp. 2535–2540, 2007.

[74] E. C. Paraiso and J. A. Earth, "An Intelligent Speech Interface for Personal Assistants in R & D Projects," pp. 804–809.

[75] "Protégé Webpage." [Online]. Available: http://protege.stanford.edu/.

[76] M. Fuckner, J.-P. Barthes, and E. E. Scalabrin, "Using a personal assistant for exploiting service interfaces," *Proc. 2014 IEEE 18th Int. Conf. Comput. Support. Coop. Work Des.*, pp. 89–94, May 2014.

[77] G. Czibula, A.-M. Guran, I. G. Czibula, and G. S. Cojocar, "IPA - An intelligent personal assistant agent for task performance support," *2009 IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, pp. 31–34, Aug. 2009.

[78] Z. Suryady, U. Sarwar, and M. Abbas, "A gateway solution for IPv6 wireless sensor networks," *2009 Int. Conf. Ultra Mod. Telecommun. Work.*, pp. 1–6, Oct. 2009.

[79] S. Campos, J. J. P. C. Rodrigues, M. L. Oliveira, L. D. P. Mendes, E. F. Nakamura, and C. Maur, "Design and Construction of a Wireless Sensor and Actuator Network Gateway based on 6LoWPAN," pp. 4–7.

[80] S. Federal, S. Campos, J. J. P. C. Rodrigues, L. D. P. Mendes, E. F. Nakamura, and C. Maur, "Design and Construction of Wireless Sensor Network Gateway with IPv4 / IPv6 Support," pp. 2–6, 2011.

[81] L. F. Schrickte, C. Montez, R. De Oliveira, and A. R. Pinto, "Integration of Wireless Sensor Networks to the Internet of Things Using a 6LoWPAN Gateway," *2013 III Brazilian Symp. Comput. Syst. Eng.*, pp. 119–124, Nov. 2013.

[82] A. Botta, W. De Donato, V. Persico, and A. Pescap, "On the Integration of Cloud Computing and Internet of Things."

[83] J. Zhou, T. Leppänen, E. Harjula, C. Yu, H. Jin, and L. T. Yang, "CloudThings : a Common Architecture for Integrating the Internet of Things with Cloud Computing," pp. 651–657, 2013.

[84] C. Dores, L. P. Reis, and N. V. Lopes, "Internet of things and cloud computing," *2014 9th Iber. Conf. Inf. Syst. Technol.*, pp. 1–4, Jun. 2014.

[85] P. R. B. B, P. Saluja, N. Sharma, A. Mittal, and S. V. Sharma, "Cloud Computing for Internet of Things & Sensing Based Applications," in *2012 Sixth International Conference on Sensing Technology*, 2012, pp. 374–380.

[86] C. Qiang, G. Quan, B. Yu, and L. Yang, "Research on Security Issues of the Internet of Things," *Int. J. Futur. Gener. Commun. Netw.*, vol. 6, no. 6, pp. 1–10, Dec. 2013.

[87] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *IEEE Comput.*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[88] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A Standardization Perspective," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 265–275, Jun. 2014.

[89] A. P. Abidoye, "Using Wearable Sensors for Remote Healthcare Monitoring System," *J. Sens. Technol.*, vol. 01, no. 02, pp. 22–28, 2011.

[90] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, "An Advanced Wireless Sensor Network for Health Monitoring."

[91] "AlertMe Webpage." [Online]. Available: http://www.alertme.com/.

[92] "Lowes Webpage." [Online]. Available: http://www.lowes.com/cd_What+Is+Iris_695688710_?cm_sp=Electrical-_-HomeAutomation|HB-_-Merch|Iris_Search_Learn_More.

[93] S. Hwang and D. Yu, "Remote Monitoring and Controlling System Based on ZigBee Networks," vol. 6, no. 3, pp. 35–42, 2012.

[94] S. Kumar, "Ubiquitous Smart Home System Using Android Application," *Int. J. Comput. Networks Commun.*, vol. 6, no. 1, pp. 33–43, 2014.

[95] W. Fu, G. Chen, P. Wang, Y. Hong, and H. Ge, "Improvement of Home Appliance Control System in Smart Home Based on 6LoWPAN," *J. Comput.*, vol. 9, no. 1, pp. 183–188, Jan. 2014.

A IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments

João Santos[1], Bruno M. C. Silva[1], Joel J. P. C. Rodrigues[1], João Casal[2], Kashif Saleem[3]

[1] Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
[2] TIMWE I &D, Parkurbis – Parque de Ciência e Tecnologia da Covilhã, Tortosendo, Portugal
[3] Center of Excellence in Information Assurance (CoEIA), King Saud University (KSU), Riyadh, Kingdom of Saudi Arabia

joao.santos@it.ubi.pt; bruno.silva@it.ubi.pt; joeljr@ieee.org; joao.casal@timwe.com; ksaleem@ksu.edu.sa

***Abstract*— The evolution of mobile devices have triggered the appearance of Intelligent Personal Assistants (IPAs). IPAs are software agents used to help users to fulfill several daily life actions. They are supposed to be intelligent in a way that allows them to give their owners advices about many different subjects. To do so, IPAs must learn what is the usual behavior of the users by interact directly with them. With the evolution on various technological areas, scenarios of ubiquitous communication can be created. One of the potential enablers for those scenarios is the Internet of Things (IoT) paradigm were machines with decision support systems interact and communicate between them. In an IoT environment, IPAs can interact with other smart objects in order to gain new knowledge and awareness about their users. This paper proposes a novel IoT-based mobile gateway solution for mobile health (m-Health) scenarios. This gateway autonomously collects information about the user/patient location, heart rate and possible fall detection. Afterwards, it forwards, in real time, the collected information to a caretaker IPA that will manage a set of actions and alarms if necessary. The algorithms used for each mobile gateway service, and the differentiation between the mobile gateway acting as a communication channel or a smart object are also addressed on this paper.**

Keywords*— *Internet of Things; Mobile Health; Intelligent Personal Assistant; Body Sensor Network; Mobile Gateway; Smart Object

1. Introduction

The use of personal assistants is not a new trend, since they are used through daily business and personal tasks for many years. Personal assistants are responsible for managing several daily tasks, such as paying bills, making appointments, shopping, taking notes, dealing with correspondence, answering phone calls, among others. Formerly, the personal assistant role was performed by persons; nowadays the personal assistant role can also be made by software agents with learning capabilities and contextual knowledge.

Intelligent Personal Assistants (IPAs) are software agents that can automate and ease many of the daily life tasks of their users [1][2]. IPAs gain knowledge and awareness about the usual behavior of their users by interacting with them and/or collecting awareness data (e.g., location and context). Therefore, IPAs are supposed to adapt themselves to their user's needs and actions, in order to improve the assistance given.

The appearance of the Internet of Things (IoT) paradigm enables the creation of scenarios of ubiquitous communication, where all addressable objects in the environment are supposed to be able to communicate between them within the network [3][4]. IoT offers new technological opportunities to create new types of services that users can benefit for both their personal and professional lives. Services where humans interact directly with machines to access other remote machines to perform some desired tasks will be available. Smart Homes [5][6], Smart Cities [7], manufacturing [8], healthcare [9], and transports and logistics [10][11] are some examples of areas on which IoT is already creating new business opportunities.

The inclusion of software agents such as IPAs on IoT scenarios can enhance those devices capabilities to gain more knowledge and awareness about their surroundings. IPAs can even better perceive the preferences of their users by learning through their interactions with other smart objects. For example, an IPA could learn the morning routine of its user by interacting with a smart object in the environment, such as a toaster, the air conditioner or the coffee machine. Gateways are often used to establish the connection between the IoT environment and the global Internet. Static gateways are used on static environments, while mobile gateways can be useful on dynamic environments with high user mobility.

The use of smart mobile devices offers numerous opportunities to create efficient healthcare services and solutions. Mobile health (m-Health) solutions use mobile devices to deliver healthcare services anytime and anywhere, surpassing geographical, temporal, and even organizational barriers. M-Health solutions address emerging problems such as, the increasing number of chronic diseases related to lifestyle, high costs of existing national health services and the need to empower patients and families to self-care [12]

This paper presents a novel mobile gateway solution for a specific ubiquitous m-Health scenario, where a Body Sensor Network (BSN) constantly monitors a person. The BSN gives the information of the location, the heart rate and possible falls of the user, being each one of these items mapped to an individual service on the mobile gateway application existent on a smartphone. The gateway is responsible for sending the information of each service to a caretaker IPA platform called AMBRO. There, the caretaker receives notifications and alerts regarding the person being monitored on the BSN that is under his care. This paper also presents the performance evaluation of the proposed mobile gateway solution through the assessment of the power that it consumes, the accuracy of the services that it integrates, and the way that it communicates with other objects on the environment.

The main contributions of this paper are the following:

- Creation of a novel IoT-based mobile gateway solution to be integrated with an IPA platform.
- The possibility to prove the feasibility of the AMBRO on multiple environments (smart home and healthcare).
- The creation of a novel remote monitoring system for a ubiquitous m-health scenario.

The remainder of this paper is organized as follows. Section II presents a literature review about IPAs, remote monitoring systems on multiple environments, and IoT gateways. Section III briefly describes the AMBRO platform, including the layers of its architecture. Section IV presents the AMBRO mobile gateway application, indicating its different services and describing the workflow of each one. Section V shows the experiments that were performed on the AMBRO mobile application and the obtained results, while Section VI presents the conclusion of the paper, addressing some work to develop in the future.

2. Related Work

IPAs can be integrated on healthcare environments as tools to assist patients, physicians or caretakers performing several tasks. For example, IPAs can be used as a medication reminder, or give real time information about the health status of monitored patients to the medical staff.

The mobile technological revolution and the advent of advance mobile operating systems have enabled the appearance and creation of already popular personal digital assistants as one of their main features. Apple's Siri [13], Google Now [14], Samsung's S Voice [15], and Microsoft's Cortana [16] are examples of wide known and popular IPAs.

In recent years, several mobile applications have been developed that work has an IPA on a healthcare context. The work presented in [17] demonstrates an example of an Android application used to manage the medication time schedule of the users. The application, called SapoMed, allows users to register different drugs to the medication intake schedule. It uses a Web service to present the prescription information about each registered medication, giving more indicators for users to choose the intake time of each medication. The SapoMed application can be seen has a mobile personal assistant since it manages and monitors all the medication intake schedule of the users, generating alerts to remind them about the time to take the medication.

SapoFit, presented in [18] is a weight control mobile application for obesity prevention. This application allows users to manage and keep track of weight in a healthier and more practical way. Based on the previous user input information, SapoFit offers several features for users control their weight, body mass index (BMI), basal metabolic rate (BMR), sports activity, and even the possibility to conduct healthy food plans based on their caloric needs and weight.

The authors in [19] propose a health monitoring system, denominated HealthPal, that interacts with peripheral devices (such as thermometers) and PDAs to create a insightful record of the patients' health status. The system is aimed to the elderly people, on which the software application installed on their PDAs and on the medical staff devices (PCs, PDAs, etc) receive alerts if some health status indicator is alarming. This mechanism is used to allow patients to auto-diagnose themselves, providing as well a tool to improve the speed on which the medical staff gives the assistance to their patients.

The solution presented in [20] considers a caretaker that is notified if a person under his care has fallen. This scenario uses a smartphone to forward the information detected by the sensors monitoring the user to an Internet enabled device belonging to the caretaker. In this case, the terminal used by the caretaker only needs to receive the information gathered at the BSN and display it to the caretaker.

In [21] is shown a platform used to monitor the health status of persons on their home environment. The system is comprised by a *(i)* wireless sensor network (WSN) inserted on the house of the user, *(ii)* an online platform to store the data gathered from sensors, and *(iii)* the end-user devices, such as PCs, mobile phones and tablets, used by the medical staff or other people trusted by the user (family, friends, neighbors, etc). The online platform is able to detect abnormal values on the health indicators of the monitored person, and generates notifications to the caretakers' end-user devices.

The importance of the WSNs to the health monitoring systems is demonstrated on [22]. As in other scenarios, body sensors (blood pressure, EKG, heart rate, among others) and environmental sensors are used to estimate the health status of the monitored person. Those two types of networks connect to a backbone system, where a database is used to store the data gathered by the sensors. The end-user devices, as PDAs, also connect to the backbone system in order to consult the data stored on the database.

In [23] is presented the project CodeBlue, which consisted on a publish/subscribe platform model, where caregivers could subscribe to vital signs data gathered by sensors monitoring persons under their care. The caregivers could access the health status information of the monitored persons on their PCs or PDAs.

It can be necessary to forward the traffic originated by sensors to external networks, such as the Internet. To make that possible, device gateways can be used. In [24] is presented an architecture for IoT scenarios, which comprise a semantic gateway able to understand different communication protocols, like the Constrained Application Protocol (CoAP), the Extensible Messaging and Presence Protocol (XMPP) and the Message Queue Telemetry Protocol (MQTT). This work shows an interesting study about the mechanisms necessary to develop a gateway interoperable in different environments. There are also presented some services used at the gateway to differentiate the incoming traffic, easing the creation of meaningful information. However, there is not detailed how this semantic gateway can be used in real life, including its capability to work on dynamic environments.

The work presented at [25] shows a Power over Ethernet (PoE) gateway used at a medical environment, where sensors are used to check the health status of the patients. This stationary gateway is deployed on a development platform similar to the Arduino, where an Ethernet connection is used to power the gateway. The gateway is able to communicate with healthcare sensors, such as ECGs and EKGs, existent in the environment over Wi-Fi, Bluetooth or IEEE 802.15.4 connections. The usage of this gateway is not possible on scenarios with mobility demands, since, in order to operate correctly, it needs to always be powered up via an Ethernet connection.

In [26], the authors presented a ubiquitous personal health surveillance and monitoring system composed by a BSN and a gateway. The gateway role on that system is performed by an Android smartphone, which receives the data gathered on the BSN and forwards it to the health database servers on the Internet. Then, the medical staff is able to properly analyze the health indicators of their patients. The gateway on this system connects with the BSN through a sink node by using Bluetooth or USB connections. However, very little is said about the data throughput and the power consumption on the network devices.

When compared with other solutions presented on the literature, the mobile gateway on this paper differs by having the ability to communicate with sensors on a BSN while deciding, at the same time, how to manage that same communication process. The communication management makes possible to control the power consumption on the system devices, as the smartphone. Besides, the monitoring services accuracy is preserved, making the AMBRO mobile gateway an application capable to use accurate and low-power health monitoring services. Also, the AMBRO mobile gateway is supposed to be used on scenarios of high mobility, creating ubiquitous communication scenarios where persons can be constantly monitored.

3. The Ambro Platform

AMBRO is a technological platform [27] that supports the integration of IPAs on ubiquitous communication environments. AMBRO was developed to assist his user on numerous ways, giving him hints and suggestions about several aspects of his daily life. By using the AMBRO personal assistant, the users can: *(i)* access Web services of their interest (find nearer restaurants, consult the weather, among others), *(ii)* consult the information about sensors they possess (see the temperature at home for example), and *(iii)* change the behavior of smart objects in the environment (turn off the air conditioner, among others). Besides these aspects, the AMBRO personal assistant is capable of learning the usual behavior of their users by interpret the actions they perform over the time with objects existent in the environment, including the personal assistant itself.

The AMBRO platform is divided into three logical layers, *i)* the IoT world, *ii)* the Intelligent cloud system, and *iii)* the User interfaces. On the IoT world layer are present all the external devices (objects and sensors) that give IPAs the information about the environment that surrounds them. The static gateway existent on this layer is responsible for managing the communication between sensors and objects in the environment and the cloud. The static gateway supports different communication protocols, such as the ZigBee and the Z-Wave. An AMBRO proprietary protocol is also supported. As for the mobile gateway, which is one of the main contributions of this paper, it connects the BSN to the AMBRO platform. It is developed for the Android operating system, and connects with the AMBRO cloud using RESTful Web services. It also allows the discovery of Bluetooth devices that can be used on services available on the mobile gateway. Sensors are used to retrieve information on the user environment. The periodicity chosen to gather information about the environment variables is carried out independently by each sensor, allowing the user to adjust it as he wants it. Objects, such as fans or light bulbs, can be operated automatically or operated through the action of the user.

The Intelligent cloud system layer is the core of the AMBRO platform, being responsible for all the logical actions concerning IPAs. It includes the usage of databases to store the data; a security and authentication module to assure the integrity of the information flowing on the system; an artificial intelligence (AI) module to allow learning mechanisms to be used on data gathered on the system; a text-to-speech module to synthesize text strings and files into spoken words; a voice recognition module to convert spoken audio into text strings; a notifications module to allow IPAs to receive notifications about events happening on the system; and a communication module to manage the interaction between the IPAs and other external entities.

On the User interfaces layer are provided the interfaces to allow the users to interact with the AMBRO system. One of them is the back-office tool that is used to manage the user profile, sensors and objects. The other interfaces correspond to the IPAs interfaces, where can be emphasized the IPA Web, which accessible in every device with internet connectivity; the IPA mobile that can be used on Android (minimum version 4.0.3) and iOS (minimum version 8.1) smartphones; and the IPA smartwatch, available on any Android Wear OS smartwatch. The IPA smartwatch is mostly used as a complement to the mobile IPA interface, since in order to be used the smartwatch needs to be paired with an Android smartphone, mostly displaying system notifications regarding sensors and objects. Figure 1 depicts the AMBRO platform architecture.

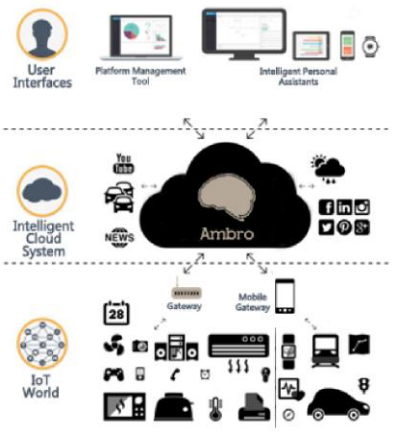

Fig. 1. AMBRO platform main architecture.

4. AMBRO Mobile Gateway

On communication scenarios where devices are constantly changing their position, it can be necessary to use a mobile gateway to forward the traffic between those devices and external networks. The use of a smartphone as a mobile gateway is one possibility to enable such scenarios.

On this paper that solution is exploited through the development of an Android application, designated AMBRO mobile gateway application. The AMBRO mobile gateway is a specific gateway application that is inserted on a remote health monitoring scenario, allowing a person that possesses an AMBRO personal assistant (the caretaker) to check some health indicators about a person under his care. In order to make the system fully functional, the monitored person must always have with him a smartphone with the AMBRO mobile gateway application installed. The AMBRO mobile gateway application allows both actors (monitored person and caretaker) to control three different monitoring services that use sensors inserted on a BSN. The BSN is used to monitor the person using the mobile gateway application. The user has to insert his AMBRO account credentials to access the AMBRO mobile gateway application. Those credentials are the same that are used by the caretaker on his AMBRO personal assistant, since in practice the mobile gateway belongs to him. Figure 2 shows the splash screen of the AMBRO mobile gateway application.

Fig.2. AMBRO mobile gateway splash screen.

If the login procedure is done correctly, the user is redirected to the application main screen. On the AMBRO mobile gateway main screen, the user has the possibility to start or stop the monitoring services. Besides that, the user is also able to see relevant information about the data gathered by each monitoring service, like the coordinates of his current location, his heart beats per minute, and the number of times he potentially fell. One other service available on the mobile gateway application is related with the search of Bluetooth objects to be used on the monitoring services. The application main screen is shown on figure 3.

Fig.3. AMBRO mobile gateway main screen.

The AMBRO mobile gateway architecture can be divided into four modules, as depicted on figure 4: *(i)* the BSN, *(ii)* the mobile gateway, *(iii)* the AMBRO cloud, and *(iv)* the IPA of the caretaker. The BSN module is composed by sensors that monitor the person who use them in real time. On this case, there are two types of devices: one Shimmer module, using an accelerometer and a gyroscope to detect if the person has fallen, and a smartwatch with an embedded heart rate sensor to count the heart beats of the person. The information gathered by both sensors is passed to the mobile gateway through a Bluetooth connection (the smartwatch uses Bluetooth Low Energy, while the Shimmer module uses Bluetooth class 2). The mobile gateway module corresponds to the smartphone that has the AMBRO mobile gateway installed, which allow the monitoring services to be controlled. One of those monitoring services (the location monitoring service) uses the GPS of the smartphone to determine accurately, and in real time, the position of the monitored person. The data gathered by the monitoring services is forwarded to the AMBRO cloud through RESTful Web services using the HTTP. Inside the AMBRO cloud, the information is stored and then processed by the multiple logical modules there existent. If some indicator about the monitored person is alarming, a notification is generated to the IPA of the caretaker through a *push* notification mechanism.

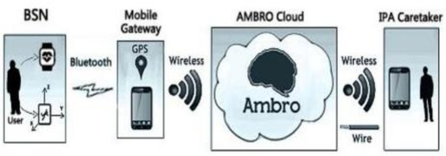

Fig. 4. AMBRO mobile gateway architecture.

4.1. Location monitoring service

The location monitoring service is a tracking service used to retrieve the position of the user through the GPS system of the smartphone. By using the low power embedded sensors of the smartphone (barometer, gyroscope, accelerometer, among others), this service is also able to discover the activity that the user is performing. This is possible by using the Google's Activity Recognition API [28], which enables the detection of multiple possible activities for the user (e.g. still, walking, running, driving a vehicle, riding a bicycle).

Other works have been published concerning tracking systems like the one being presented on the AMBRO mobile gateway. For example, on [29] the authors presented a location based application for Android, capable of changing the profile on the smartphone accordingly to the user location; locating a person through a SMS; and locating people on nearby places to the user On [30] is proposed a vehicle tracking system using the GPS system of an Android smartphone. The application can be useful to monitor the driver performance, check the speed at which a vehicle circulates, or increase the security and performance of fleet management businesses. The authors in [31] propose a smartphone tracking system used to retrieve children's current location. The application is designed to help parents track their children in real time, by sending SMS messages to the children smartphones. The smartphones of the children answer with their current location signaled on a map. On [32] a mobile tracking application using the GPS system of an Android smartphone is proposed. The application has the purpose of tracking children through their smartphones, defining geographical areas that cannot be surpassed. The parents are warned through a SMS message if their child is outside the geographical area allowed.

Regarding the location monitoring service of the AMBRO mobile gateway application, it can only be activated if the Wi-Fi and the GPS have been previously enabled on the smartphone. When this is assured, the location monitoring service is initiated and activity updates are requested every 20 seconds. If the low power embedded sensors on the smartphone detect the user is moving, GPS is turned on and location updates are requested. Location updates are stopped when the smartphone detects the user is stopped during one straight minute. This mechanism is important to diminish the power consumption on the smartphone, since the GPS is only used when the user is moving.

The location monitoring service allows data (coordinates and activity detected) to be stored on the smartphone memory if the Internet connection fails. The smartphone retries to send the data to the AMBRO cloud every time an activity detection update is requested, assuring that no location updates are lost. If the coordinates sent by the mobile gateway to the AMBRO cloud return a place outside a geofence defined dynamically by the system, the caretaker is notified by receiving an alert message on his IPA.

4.2. Heart rate monitoring service

The heart rate monitoring service is used to check the heart rate of its user. To do so, an optical heart rate sensor embedded on an Android Wear smartwatch is used. The use of an Android Wear smartwatch, in this case a Moto 360, eased the development of the service, since the manner at which the applications are built on the Android Wear OS is very similar to the Android OS. Besides, the use of an Android Wear wearable device facilitated the communication between the smartphone and the smartwatch, eliminating one of the problems inherent to the IoT networks: the communication between heterogeneous devices [33].

To start the service, the Bluetooth needs to be enabled on both the smartphone and the smartwatch. The Wi-Fi also needs to be enabled on the smartphone. If this is all assured, the service is started with the smartphone trying to establish a Bluetooth Low Energy (BLE) connection with the smartwatch. When the connection is established, the smartphone starts a mechanism to detect what is the activity that the user is performing every 20 seconds by using the Google's Activity Recognition API. This mechanism is important to detect if the heart rate values of the user are normal according to the activity performed by him. On the smartwatch the heart rate of the user is detected through the optical heart rate sensor every second. This information is sent to the smartphone through the BLE connection previously established, using the Wearable Message API [34]. By using this API, it was possible to implement a wearable listener on the smartphone to make it aware of every message sent by the smartwatch. The Wearable Message API mechanism is depicted on figure 5.

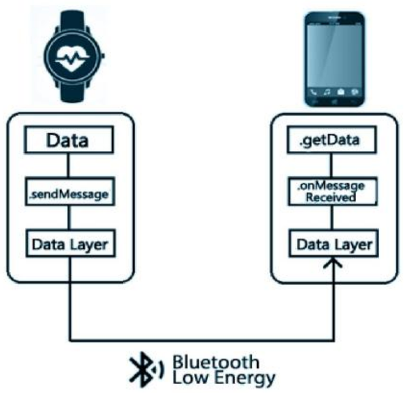

Fig.5. Wearable Message API used on the communication between the smartwatch and the smartphone on the Heart rate monitoring service.

The heart rate values sent to the smartphone (serving as mobile gateway) are summed up, and a mean of the user heart rate on the past five minutes is calculated. The gateway tries to send information to the AMBRO cloud every five minutes, containing the mean of the heartbeat and the activity the user performed more during that period.

On the AMBRO cloud there are two different typical mean values for the user: when he is stopped, and when he is in movement. Those values are obtained through the use of AI algorithms. If the heartbeat values sent to the AMBRO platform are too low or too high when compared with the typical heartbeat mean values of the user for that activity, a notification is sent to the IPA of the caretaker alerting him about the situation. Therefore, the caretaker is informed of possible health problems on the person under his care by evaluating odd heartbeat values detected by the heart rate monitoring service. If the data is not successfully delivered to the cloud, the mobile gateway stores the information on its memory to send it later. This is performed every 5 minutes.

4.3. Fall detection service

The fall detection service is used to detect if the user has fallen. To detect a fall, one of the existing possibilities is to use tri-axis accelerometers. This mechanism allows an accelerometer attached to the body of a person to measure the acceleration of the person relatively to the gravity, using the equation in (1).

$$|A| = \frac{\sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2}}{9.8} \qquad (1)$$

The use of gyroscopes can improve the fall detection systems precision. Gyroscopes can detect the orientation of a person by calculating his angular velocity through the equation (2).

$$|\omega| = \sqrt{(\omega_x)^2 + (\omega_y)^2 + (\omega_z)^2} \qquad (2)$$

On both situations, the values are compared with thresholds pre-defined by the user or dynamically defined by the system. There are some work published that use this methodology to detect falls. On the work presented in [35] is tested a fall detection system using a tri-axial accelerometer and a tri-axis gyroscope. By collecting data through some experiments in multiple daily life activities, the authors defined upper and lower thresholds for both acceleration and angular velocity. Then, using these thresholds, falls could be detected through acceleration and angular velocity values measured in real-time. In [36], the authors used the 3-axis accelerometer and magnetometer of a smartphone to detect falls. One more time, thresholds were defined for both accelerometer and magnetometer data, defining an algorithm able to detect falls when certain values below lower thresholds and values above upper thresholds were obtained. The authors of [37] demonstrated a real-time wireless fall detection system using a 3-axis accelerometer, capable of differentiating falls from typical activities from daily life.

To develop the fall detection system on the AMBRO mobile gateway, there were done some previous experiments with a Shimmer module (containing an accelerometer and a gyroscope) to assess what were the typical values for the acceleration and angular velocity when the user was performing different daily life activities. These values are shown on Table 1. This allowed the definition of appropriated acceleration and angular velocity thresholds.

Table 1.
Acceleration and angular velocity values for different daily life activities.

Activity Performed	**Physical Measures**	
	Acceleration (estimate) [G]	***Angular Velocity (estimate) [degrees/second]***
Still	0.9-1.15	0-90
Fall	0.05-3.5	150-300
Walk	0.4-1.7	0-100
Sit Down	0.3-2.4	30-140

Stand Up	0.6-1.4	30-140
Lay Down	0.5-1.5	80-200
Run	0.2-2.9	15-170
Jump	0.4-3	80-170

As the Table 1 shows, when a fall is detected the acceleration can vary from 0.05 G to 3.5 G. The lower acceleration value is detected when a person is at free fall. The higher acceleration value is usually correspondent to the instant the person hits the floor. The correct combination of the Lower Acceleration Threshold (LAT), the Upper Acceleration Threshold (UAT), and the Angular Velocity Threshold (AVT) can form a trustworthy mechanism to detect a fall. To do so, there were defined three reasonable values for each threshold: 0.3G to the LAT, 2.5G to the UAT, and 220º to the AVT. The definition of these thresholds tried to optimize the fall detection mechanism, avoiding the confusion between a fall and other daily life activities, especially a run or a jump.

To activate the fall detection service on the AMBRO mobile gateway, the user has to choose the Shimmer sensor that he wants to use on a first place. After that, the mobile gateway has to establish a Bluetooth connection with the Shimmer sensor module that was chosen. The Shimmer module starts streaming every 20 milliseconds, gathering values from the accelerometer and from the gyroscope. A fall is detected every time the UAT is exceeded, and in the next 0.8 seconds, at least three values below the LAT are detected. If this premise is fulfilled, it is possible for a person to be at a free fall. To confirm that a person is effectively falling, it is needed to ascertain if in the next 1.5 seconds, the acceleration surpasses again the UAT, and the angular velocity exceeds the AVT. If all this is true, one last step needs to be performed. The mean value for the acceleration in the next two seconds needs to be less than 1.2 G, indicating that the person is still, and possibly lying on the floor. This indicates that a fall has been detected. The AMBRO mobile gateway has to send a notification to the IPA of the caretaker to inform about the situation. If the notification fails to be delivered, the mobile gateway will try to resend it one minute later. This process will repeat until the notification has been delivered or the fall detection service has been stopped by the user.

4.4.Object search service

The object search service is used to discover sensors that communicate over Bluetooth, like the Shimmer devices. The Bluetooth has to be enabled on the smartphone. The system will enable the Bluetooth adapter automatically on the smartphone if it is not enabled when the service tries to start.

Firstly, the gateway sends a broadcast to the environment to search Bluetooth devices. When the discovery process is over, a list of Bluetooth devices that answered positively to the broadcast message is returned to the user. Then, the user can select one of those devices to be paired with the gateway. If the device that has been paired with the gateway is a Shimmer sensor, it will be eligible to be used on the fall detection service. Figure 6 shows a list of devices returned by the object search service on the AMBRO mobile gateway.

Fig.6. List of devices returned by the object search service on the AMBRO mobile gateway.

The object search service can only be launched if the heart rate monitoring service and the fall detection service are not running. This mechanism prevents the degradation of the communication between the mobile gateway and other Bluetooth devices, since it does not allow the communication environment to be burdened by Bluetooth broadcast messages.

5. Results Analysis

There were defined three metrics to assess the performance of the AMBRO mobile gateway: *i)* power consumption of the devices; *ii)* accuracy of each monitoring service available at the mobile gateway; and *iii)* interoperability with other objects existent on the environment.

5.1.Power Consumption

The power consumption of devices in IoT environments is very important to evaluate the system performance on such scenarios [38]. This is due to the characteristics that IoT devices present, such as high mobility, limited power supply, and capacity to communicate anytime and anywhere. To respect all these conditions without degrade the system performance some strategies need to be applied. One of those strategies is to only make objects communicate when is strictly necessary. By doing this, the quick discharge of the devices' batteries is avoided. Different experiments were conducted on the AMBRO mobile gateway application to evaluate the parameters that were changeable to make the power consumption of the system optimized. For both the location monitoring and heart rate monitoring services different experiments were performed by simply varying the rate on which activity detection updates were requested. Another parameter which varied for each experiment performed on the heart rate monitoring service was the periodicity at which the data was sent to the AMBRO cloud. On the fall detection service the reading frequency for the acceleration and for the angular velocity on the Shimmer module varied for each scenario where the AMBRO mobile gateway was tested. To perform the experiments, it was used a Samsung Galaxy Express 2, equipped with the Android OS 4.4.2, CPU dual-core 1.7 GHz, internal memory up to 8 GB , 1.5 GB RAM, BLE connectivity, and Wi-Fi, 2G, 3G, 4G and LTE wireless network connectivity to perform the mobile gateway role. It was also used a Moto 360 smartwatch, equipped with the Android Wear OS (version 5.0.2), a Texas Instruments OMAP 3 processor, 4 GB of internal memory and 512 MB RAM to measure the heart rate of the monitored person. It was necessary to verify how many hours the battery of the smartphone and the battery of the smartwatch could handle when performing normal daily life tasks, in order to evaluate correctly the results returned for each experiment scenario. So, it was ascertained that both smartphone and smartwatch could handle, at average, 40 hours without discharging completely their batteries. Table 2 presents the first experiment scenario performed to evaluate the power consumption of the devices existing on the AMBRO mobile gateway communication environment.

Table 2.
AMBRO mobile gateway first experiment scenario.

Services	**Tested Features**					
	Activity Detection	***Check HR***	***Reading Frequency***	***Send Data to the Cloud***	***Smartwatch Battery***	***Smartphone battery***
Location Service	20 sec	------	-----	When a location update is requested		
HR Service	20 sec	1 sec	-----	5 minutes	11h 15mins	19h (approx.)
Fall Detection Service	----	----	20 msec	When a fall is detected		

On the first experiment scenario the activity of the user on the location monitoring service and on the heart rate monitoring service was checked with a periodicity of 20 seconds. The heart rate of the monitored person was measured every second, with that information being sent to the AMBRO cloud every 5 minutes. The Shimmer module operated on a 20 milliseconds rate to detect if the user has fallen. On the location monitoring service the communication with the AMBRO cloud was not performed periodically, since location updates are not requested if the user was not in movement. The fall detection service only generated notifications when a fall was detected, and only on those situations the communication with the cloud was performed. On this experiment scenario was concluded that the smartwatch battery lasted 11 hours and 15 minutes and the smartphone battery lasted 19 hours approximately. On Table 3 is presented the second experiment scenario to evaluate the power consumption of the AMBRO mobile gateway application.

Table 3.
AMBRO mobile gateway second experiment scenario.

Services	**Tested Features**					
	Activity Detection	***Check HR***	***Reading Frequency***	***Send Data to the Cloud***	***Smartwatch Battery***	***Smartphone battery***
Location Service	40 sec	------	-----	When a location update is requested		
HR Service	40 sec	1 sec	-----	5 minutes	11h 15mins	19h 15mins (approx.)
Fall Detection Service	----	----	50 msec	When a fall is detected		

There can be noticed some changes on some experiment parameters when comparing the first experiment scenario with the second experiment scenario. Essentially, the activity detection rate on the second experiment scenario was twice when compared with the first experiment scenario. By doing this, the smartphone battery could be a little spared. On the fall detection service, the Shimmer module read the acceleration and the angular velocity at a 50 milliseconds rate. The heart rate of the user was checked every second as in the first experiment scenario. So, the smartwatch battery lifetime on the both experiment scenarios was the same. On this second experiment scenario the battery of the smartphone lasted 19 hours and 15 minutes, approximately. On Table 4 is presented the last experiment scenario performed on the AMBRO mobile gateway application environment to assess the power consumption on the devices.

Table 4.
AMBRO mobile gateway third experiment scenario.

Services	**Tested Features**					
	Activity Detection	***Check HR***	***Reading Frequency***	***Send Data to the Cloud***	***Smartwatch Battery***	***Smartphone battery***
Location Service	60 sec	------	-----	When a location update is requested		
HR Service	60 sec	1 sec	-----	5 minutes	11h 15mins	19h 30mins (approx.)
Fall Detection Service	----	----	100 msec	When a fall is detected		

On the third experiment scenario the activity detection was performed every minute. The Shimmer module analyzed the acceleration and the angular velocity every 100 milliseconds. The information relative to the heart rate monitoring service was delivered to the cloud every 10 minutes. These changes allowed the smartphone battery to last 19 hours and 30 minutes. (more 30 minutes than the first experiment scenario, and more 15 minutes than the second experiment scenario).

Although the variation of the parameters on the third experiment scenario allowed the smartphone battery to last more than on the other experiment scenarios, the precision of the services of the system in this case is also lower. In the activity detection scenario, as the user activity is detected less often, it is more likely that some user movements are not detected. It generates less location updates, resulting on a system performance degradation. On the fall detection system, the probability of a fall not being detected increases as less frequent the acceleration and angular velocity values are checked by the Shimmer module. This is due to the time interval on which a fall starts to be noticed being as small as possible, since a fall is originated by a sudden movement in the order of milliseconds..

The chosen solution to be implemented on the AMBRO mobile gateway application was the one evaluated on the first experiment scenario, due to the better commitment between the power consumption of the battery on the devices (of the smartphone and the smartwatch) and the overall performance of the system. By doing this, it was assured that no important information was lost during the operation of each service available at the AMBRO mobile gateway application, while the power used to make those services operate at the chosen conditions was not too prejudicial to the system itself.

5.2.Accuracy of the monitoring services

By evaluating the accuracy of the results returned by the monitoring services of the AMBRO mobile gateway application it is possible to comprehend if it is conducting his job properly. The location monitoring service accuracy can be obtained by evaluating the operating mode of the mechanisms used to detect the user activity and the GPS coordinates. The activity detection mechanism on the AMBRO mobile gateway application was used to detect if the user was quiet in some place or if he was in movement. If the user was still and not using his smartphone for any purpose, the activity detection mechanism always returns that the user is still. If the user was walking or driving a car, the activity detection mechanism detects that the user is in movement, requesting location updates through the most accurate location provider available at the moment (network, power cells or GPS). The location update mechanism is more or less accurate depending on the location provider that is used to get the GPS coordinates. For example, if the location update mechanism is capable of using the GPS provider to get coordinates, the accuracy can vary from 10 to 30 meters; if the location update mechanism is only capable of using the power cells to get user location, the accuracy can drop to distances in order of 2000 meters. To test the accuracy of the location monitoring service, there were collected 1800 GPS coordinates samples when the user was outdoor and another 1800 GPS coordinates samples when the user was indoor. Table 5 shows the accuracy of the location monitoring service when the user was indoor and when the user was outdoor.

Table 5.
Accuracy of the GPS location monitoring service.

Location	Range (in meters)					
	<50m	*50-200m*	*200-500m*	*500-1000m*	*1000-2000m*	*>2000m*
Outdoor	91%	7%	2%	--------	--------	----------
Indoor	15%	15%	15%	17%	21%	17%

GPS coordinates are more accurate in outdoor locations with 91% of the gathered samples having accuracy better than 50 meters. In comparison, at indoor locations, only 15% of the samples have accuracy better than 50 meters. These results were already expected, since at outdoor locations the smartphone is more able to get the location coordinates through the GPS system, while at indoor locations the coordinates are usually gathered through the network.

The results returned by the heart rate monitoring service are dependent on how accurate is the optical heart rate sensor embedded in the smartwatch. The accuracy of those types of sensors is good when the person who uses it is still, and deteriorates when the person is in movement. As the optical heart rate sensor embedded in the smartwatch sometimes returns heartbeat values equal to zero, the service cannot be used to detect if a person faints or collapses since it could detect too many false positive values. Instead, the service evaluates what is the usual heartbeat when the person is still and the usual heartbeat when a person is moving.

Table 6 shows the percentage of non-zero values for scenarios where the person using the smartwatch was still, and other scenarios when the smartwatch was used while the person was moving. For each situation were gathered 18000 samples.

Table 6.
Accuracy of the heart rate monitoring service.

Activity	HR Samples and respective accuracy			
	Number of Samples	*Zero Values*	*Non Zero Values*	*Accuracy (%)*
Still	18000	2340	15660	87%
In Movement	18000	4320	13680	76%

Analyzing the results shown on Table 6, while the user was stopped the optical heart rate sensor of the Moto 360 returned values different than zero around 87% of the time. When the user was moving the results only shown valid values about 76% of the time. As expected, this can be related to the displacement of the smartwatch in the wrist of the user while he is moving.

The fall detection service must only detect a fall when a person effectively falls. But sometimes, when a person is performing some daily life activities, the algorithm used by the accelerometer and the gyroscope can fail and detect falls when they did not occur. Two examples of daily life activities that can originate false positive results are running and jumping, since the thresholds used to detect falls can be sometimes surpassed. To test the accuracy of the fall detection service, it was ascertained how many falls the Shimmer module detected when its user was performing different daily life activities. Table 7 shows the results of the experiments performed to evaluate the accuracy of the fall detection service.

Table 7.
Accuracy of the fall detection service.

Activities Performed	**Falls detected when performing different daily life activities**		
	Number of tests	***Falls detected***	***Algorithm accuracy(%)***
Still	30	0	100%
Walk	30	0	100%
Sit down	30	0	100%
Stand up	30	0	100%
Lay down	30	0	100%
Jump	30	1	96.7%
Run	30	4	86.7%
Fall	30	28	93.3%

The analysis of the Table 7 allows the identification of two situations where there were detected falls when they did not occur: when the person was jumping, and when the user was running. On the 30 experiments performed while the user was jumping, the Shimmer module identified, incorrectly, 1 fall, originating an accuracy of 96.7%. When the user was running there were incorrectly detected 4 situations identifying a possible fall of the user, originating an 86.7% accuracy. On the 30 experiments on which the user was effectively falling, 28 of them were correctly identified, giving 93.3% accuracy for the service.

5.3.Interoperability with other objects

The AMBRO mobile gateway is able to search other objects in the environment through Bluetooth broadcast messages. This happens when the object search service is used. Searching Bluetooth objects in the environment is a process that can cause a high battery drain and should not be performed very often. Despite that, by using sensors from different vendors, like the embedded heart rate sensor from the Motorola smartwatch, the Shimmer sensors, and the embedded sensors from the smartphone that acts as gateway itself, the AMBRO mobile gateway uses a very important concept in IoT environments, the interoperability between objects [33][4]. One way to improve the overall performance of the system could be the use of BLE sensors instead of Bluetooth sensors, since BLE allows sensors to announce what services they can offer to other objects through a client-server model.

6. Conclusions and Future Work

AMBRO is an IPA platform able to be included in ubiquitous communication scenarios. It is supposed to help their owners on many of their daily life tasks, acting on many of those situations in an autonomous way. The architecture of the IPA platform defines the existence of several user interfaces that AMBRO users can use, a cloud infrastructure that allows the processing of many mechanisms of the AMBRO platform, and a external environment comprising several intelligent objects that AMBRO can interact with. In this last architecture layer, gateways are needed to forward the traffic between the objects. Static gateways can be used for static environments (smart home, for example). Mobile gateways are used for environments where objects are constantly moving (mobile health scenarios), like the scenario explored on this paper. This scenario comprised the use of a Shimmer sensor module to detect falls, an optical heart rate sensor embedded on a smartwatch to detect the heart rate of its user, and a GPS sensor included on a smartphone to detect the location of the user. All the information gathered by those sensors is important to caretakers using the AMBRO personal assistant, since it allows them to check the health status of persons under their care anytime and anywhere.

The development of the AMBRO mobile gateway application allowed the creation of a ubiquitous communication scenario where heterogeneous devices (the Samsung smartphone, the Motorola smartwatch and the Shimmer sensor device) can communicate. Besides that it also helped to prove that the AMBRO personal assistant can be used on different types of environments. However, some work can be performed in the future in order to improve the mobile gateway performance. One possible improvement would be the use of sensors communicating through Bluetooth Low Energy, since it would allow those sensors to announce what services they could offer to the gateway, diminishing at the same time the power consumption needed on the communication process. Other improvements that can be enumerated can be associated to: the development of the mobile gateway application to other mobile operating systems, such as the iOS; the development of new communication mechanisms to make the mobile gateway communicate through protocols that use less energy, like the Constrained Application Protocol (CoAP); the inclusion of more sensors on the BSN, increasing the number of vital signs monitored on the patients.

Acknowledgments

This work has been partially supported by TIMWE *Investigação e Desenvolvimento*, through the AMBRO project, co-funded by COMPETE/QREN/EU, in the context of the *Portuguese Sistema de Incentivos à I&DT* (project code:33779), by Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project.

References

[1] G. Czibula, A.-M. Guran, I. G. Czibula, and G. S. Cojocar, "IPA - An intelligent personal assistant agent for task performance support," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 31–34.

[2] S. Pais, J. Casal, R. Ponciano, and S. Lourenço, "Unsupervised Assistive and Adaptive Intelligent Agent in Smart Environment," in *Conference: ICIES 2015: International Conference on Intelligent Environments and Systems, Paris, France, (Jan 23-24, 2015), At Paris, France*, 2015.

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[4] L. Tan and N. Wang, "Future internet: The Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, pp. 376–380.

[5] K. Bing, L. Fu, Y. Zhuo, and L. Yanlei, "Design of an Internet of Things-based smart home system," *2011 2nd Int. Conf. Intell. Control Inf. Process.*, pp. 921–924, Jul. 2011.

[6] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart Home: Integrating Internet of Things with Web Services and Cloud Computing," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, 2013, pp. 317–320.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[8] Z. Bi, S. Member, L. Da Xu, and C. Wang, "Internet of Things for Enterprise Systems of Modern Manufacturing," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.

[9] D. Singh, U. S. Tiwary, and W.-Y. Chung, "IP-based ubiquitous healthcare system," in *2008 International Conference on Control, Automation and Systems*, 2008, pp. 131–136.

[10] W. Liu and Z. Gao, "Study on IOT based Architecture of Logistics Service Supply," *Int. J. Grid Distrib. Comput.*, vol. 7, no. 1, pp. 169–178, 2014.

[11] A. Menon, R. Sinha, D. Ediga, I. Technology, and O. A. Road, "Implementation of Internet of Things in Bus Transport System of Singapore," *Asian J. Eng. Res.*, 2013.

[12] B. M. C. Silva, J. J. P. C. Rodrigues, I. de la Torre Díez, M. López-Coronado, and K. Saleem, "Mobile-health: A review of current state in 2015.," *J. Biomed. Inform.*, vol. 56, pp. 265–272, Jun. 2015.

[13] "Apple Siri Webpage." [Online]. Available: https://www.apple.com/ios/siri/.

[14] "Google Now Webpage." [Online]. Available: http://www.google.com/landing/now/.

[15] "Samsung S Voice Webpage." [Online]. Available: http://www.samsung.com/global/galaxys3/svoice.html.

[16] "Microsoft Cortana Webpage." [Online]. Available: http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana.

[17] B. M. Silva, I. M. Lopes, M. B. Marques, J. J. P. C. Rodrigues, and M. L. P. Jr, "A Mobile Health Application for Outpatients Medication Management," in *IEEE International Conference on Communications*, 2013, pp. 4389–4393.

[18] J. J. P. C. Rodrigues, I. M. C. Lopes, B. M. C. Silva, and I. D. La Torre, "A new mobile ubiquitous computing application to control obesity: SapoFit.," *Inform. Health Soc. Care*, vol. 38, no. 1, pp. 37–53, Jan. 2013.

[19] A. Komninos and S. Stamou, "HealthPal : An Intelligent Personal Medical Assistant for Supporting the Self-Monitoring of Healthcare in the Ageing Society," in *4th International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2006.

[20] E. T. Horta, I. C. Lopes, J. J. P. C. Rodrigues, and S. Misra, "Real Time Falls Prevention and Detection with Biofeedback Monitoring Solution for Mobile Environments," *2013 IEEE 15th Int. Conf. e-Health Networking, Appl. Serv. (Healthcom 2013)*, no. Healthcom, pp. 594–600, Oct. 2013.

[21] A. Schrader, D. Carlson, and P. Rothenpieler, "SmartAssist - Wireless Sensor Networks for Unobtrusive Health Monitoring," pp. 84–89, 2010.

[22] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, "An Advanced Wireless Sensor Network for Health Monitoring," in *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, 2006, pp. 2–5.

[23] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care," in *Wearable and Implantable Body Sensor Networks*, 2004.

[24] P. Desai, A. Sheth, and P. Anantharam, "Semantic Gateway as a Service architecture for IoT Interoperability."

[25] J. Granados, A.-M. Rahmani, P. Nikander, P. Liljeberg, and H. Tenhunen, "Towards Energy-Efficient HealthCare: an Internet-of-Things Architecture Using Intelligent Gateways," in *Proceedings of the 4th International Conference on Wireless Mobile*

Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies," 2014.

[26] C.-H. Wang, "UPHSM: Ubiquitous personal health surveillance and management system via WSN agent on open source smartphone," *2011 IEEE 13th Int. Conf. e-Health Networking, Appl. Serv.*, pp. 60–63, Jun. 2011.

[27] P. Santos, L. Varandas, T. Alves, C. Romeiro, J. Casal, S. Lourenço, and J. Santos, "A Pervasive System Architecture for Smart Environments in Internet of Things Context," in *ICMI 2015 : XIII International Conference on Multimodal Interaction, London, UK*, 2015.

[28] "Activity Recognition API Webpage." [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi.

[29] S. Vanjire, U. Kanchan, G. Shitole, and P. Patil, "Location Based Services on Smart Phone through the Android Application," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 1, pp. 4982–4987, 2014.

[30] A. Chavan, S. Kamble, A. Bhagyshree, R. Sonawane, and R. C. Gadri, "Land Vehicle Tracking System Using Java on Android Platform," *Comput. Eng. Intell. Syst.*, vol. 3, no. 5, pp. 88–94, 2012.

[31] A. Al-Mazloum, E. Omer, and M. F. A. Abdullah, "GPS and SMS-Based Child Tracking System Using Smart Phone," *Int. J. Electr. Comput. Electron. Commun. Eng.*, vol. 7, no. 2, pp. 171–174, 2013.

[32] R. Kinage, J. Kumari, P. Zalke, and M. Kulkarni, "Mobile Tracking Application," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 2, no. 3, pp. 617–623, 2013.

[33] A. J. Jara, L. Ladid, and A. Skarmeta, "The Internet of Everything through IPv6 : An Analysis of Challenges , Solutions and Opportunities," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 4, no. 3, pp. 97–118, 2013.

[34] "Android Wear Wearable Message API." [Online]. Available: https://developer.android.com/training/wearables/data-layer/messages.html.

[35] Q. T. Huynh, U. D. Nguyen, S. V Tran, A. Nabili, and B. Q. Tran, "Fall Detection System Using Combination Accelerometer and Gyroscope," in *Proceedings of the Second International Confernece on Advances in Electronic Devices and Circuits - EDC 2013*, 2013, vol. 3200, pp. 978–981.

[36] S. Hwang, M. Ryu, Y. Yang, and N. Lee, "Fall Detection with Three-Axis Accelerometer and Magnetometer in a Smartphone," in *International Conference on Computer Science and Technology , CST 2012*, 2012, pp. 65–70.

[37] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang, "A Wearable Wireless Fall Detection System with Accelerators," in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, no. 2, pp. 2259–2263.

[38] E. Lee and C. Kim, "An Intelligent Green Service in Internet of Things," *J. Converg.*, vol. 5, no. 3, pp. 4–8, 2014.