A Middleware for Intelligent Environments in Ambient Assisted Living

R. Pereira, C. Barros, S. Pereira, P. M. Mendes and C. A. Silva

Abstract— **The increasing elderly population is changing the demography of many countries, becoming a major issue for society. As a direct consequence of this demographic phenomenon, allied to technological developments and pressure to reduce healthcare costs, new healthcare technologies for proactive health and elder care is needed. However, interoperability issues among different levels of available technologies restricts a wider deployment among intermediate and end-users. This work presents a service-oriented middleware that was developed to provide access to the functionality offered by virtually any existing device, or application, in a residential setting, in a transparent and intuitive form. The proposed middleware abstracts the communication technologies involved, allowing to seemingly integrate different communication protocols, making possible the distributed cooperation of devices that were built as stand-alone products. As a test case, an application was built with the middleware to monitor blood pressure and electromyography signal using typical health devices.**

I. Introduction

Technological advances in health care allow people to live longer and better. Prolonging life is a great achievement but the consequence is to increase the elderly people proportion, which will dramatically raise the demand for the daily need to support and monitor senior citizens. From a social standpoint, it is important that all people requiring support in their daily activities remain integrated in society, regardless of their age. However, this integration has enormous economic implications, namely those related with health systems [1].

This reality is leading to a paradigm shift in healthcare regarding patients monitoring. It′s necessary to replace centralized and reactive models of health systems (focused primarily on the disease) by distributed, and proactive, systems (focused mainly in welfare management). In this context, the distributed and continuous control of health is a key element in the transition to more proactive and accessible health systems. These distributed systems allow for an individual to monitor closely any changes in its vital signals, in order to help him maintain a desirable state of health. Furthermore, when integrated in an e-Health system, it may alert the medical staff, whenever changes occur that require medical intervention [2], [3].

In the current landscape, we have a wide range of different health devices, such as blood pressure monitors, glucose meters, electrocardiographs or electromyographs, developed by different manufacturers. Many of them permit to store information in a computer using an in-house application. There is, also, a wide range of communication standards, such as Bluetooth, Wi-Fi or RS-232/RS-485. This wide diversity of manufacturers and communication standards, although desirable, make the integration of these devices hard to attain.

The middleware proposed in this paper aims to address the integration of independent health devices, which use different communication standards, in a distributed and cooperative health system. This infrastructure can be particularly important given that the current alternatives to this technology involves the use of several devices that require manual recording and are not compatible with each other, making it difficult to analyze several variables of the patient and to store it in a database. Moreover, this infrastructure, being an intelligent monitoring system, allows sending alerts when the parameters exceed the thresholds previously established. Also, by defining a clear API, it is possible to integrate new modules for collecting other relevant biological signals.

The remainder of the paper is organized as follows. In Section II, we review several related works and identify their main contribution. The architecture, the services and the elements of the network are developed in Section III. In Section IV, we present an implementation using our middleware. Finally, in Section V, we discuss the main conclusions and future work.

II. Related Work

There are other research projects exploring medical sensor networks. Most of these projects are concerned with developing wearable medical sensors, while others have developed infrastructures for monitoring individual patients during daily activities, at home or at a hospital. In contrast, our focus is to develop a robust, scalable infrastructure for deploying sensor networks in a range of medical settings.

The Codeblue project aims to provide a wireless infrastructure for various healthcare settings, with application in a hospital environment. The main features of this technology are the creation of ad-hoc networks, routing, security and authentication control as well as filtering and storing vital signals data. The application for real time sorting and the user interface are located in a PDA [4].

The MIThril platform uses portable sensors for monitoring the user physiological state and the surrounding environment, in order to discover new techniques for human-computer interaction. The sensors, worn on the body, extract relevant features and use this data to model the user [5].

The HealthGear is a wireless sensors network system connected via Bluetooth to a mobile phone that stores, transmits and analyses the physiological data, presenting them to the user. The research is focused on using the system to recognize patterns of human behaviour [6].

The authors are with Electronics Department, University of Minho, Campus Azurém, Guimarães, Portugal. Corresponding author: csilva@dei.uminho.pt

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

The AMON project integrates a number of biosensors in a monitoring pulse device, such as heart rate, electrocardiogram, blood pressure, oximetry, skin breathing and body temperature . Moreover, it transmits the data to a remote telemedicine center for posterior analysis and, if necessary, possible emergency care [7].

The AlarmNet is defined as a home monitoring and assisted living network. This system unifies and accommodates heterogeneous devices on a common architecture that includes wireless sensors for physiological or environment data [8].

The MobiHealth project provides a complete mobile platform for monitoring patients through a wireless body sensor network (using technologies such as Bluetooth or ZigBee). The data collected by the sensors is stored in a network central point where a medical team is responsible for monitoring [9].

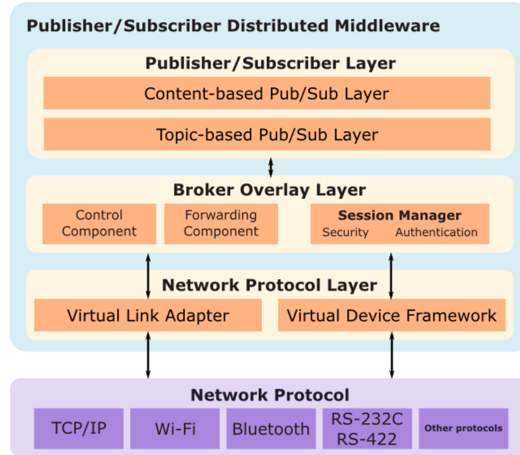

Fig. 1. Middleware Architecture.

III. Technological Infrastructure

The proposed communication infrastructure is designed to provide flexibility, adaptability and security for distributed monitoring of health devices through other devices, such as Smartphones, Tablets or Notebooks in a range of health settings. Since this environment is dynamically changing the connection among them, asynchronous interconnections are particularly useful in this context. A common approach to achieve decoupling of interacting components is by using an event-notification or publish-subscribe pattern [10], [11]. We have adopted this approach in this work.

A distributed system implemented using our middleware is composed by three types of components: raw event producers, event clients and event brokers. Raw event producers are health devices that produce data in the form of raw samples, as in a blood glucose meter or in the form of a raw stream of samples as in an electrocardiograph. Event clients can be either publishers or subscribers. They use the services provided by the middleware to communicate through events. The event brokers represent the actual middleware, providing a distributed implementation of the functionality required by the event clients.

In the next section, we introduce the architecture of our middleware and discuss each layer.

A. Middleware Architecture

The architectural model of the middleware is composed of three logical layers, namely, the Network Protocol Layer, the Broker Overlay Layer and the Publisher/Subscriber Layer - Fig. 1.

The Network Protocol Layer is composed of two components: the Virtual Link Adapter and the Virtual Device Framework. The Virtual Link Adapter permits that different communication protocols may be integrated, providing a virtual communication link to the upper layers. It also allows that the Broker Overlay Layer may route events through different communication protocols, e.g., Bluetooth to Wi-Fi. On the other hand, the Virtual Device Framework was designed to abstract the variability in the interaction with medical devices. Each medical device has a proprietary transaction protocol that is utilized to interact with the device to extract information. This transaction is specific and may use different communication protocols, e.g., Bluetooth or Wi-Fi. This component permits to adapt the proprietary protocol, creating a virtual link between the upper layers and the device. For each new supported device, a new adapter has to be written.

The Broker Overlay Layer provides the underlying distributed event routing. The Broker is implemented in terms of three components: control, forwarding and session manager. The control component cooperates with its peer brokers on other hosts to build and maintain the broker network. The forwarding component routes events over the broker network. Both the control and forward components share a state component; this encapsulates key states of the broker, such as connected clients list (publishers and subscribers), or forwarding clients list. Finally, we have the session manager. This component receives a subscription request and then initiates an authentication with the proxy server through the broker overlay. Besides, the session manager controls, also, the closing of the session upon an unsubscribe request with the proxy server. The role of the forwarding component and the proxy server is developed below.

Publish/subscribe systems can generally be classified into two classes [10]: topic-based and content-based. In a topic-based system, the subscriber specifies a topic of his interest, which may be provided by any publisher. In a content-based system, the subscriber may specify a filter on the topic; only the messages that satisfy the filter are delivered. We implemented both types of approaches in the Publisher/Subscriber Layer. Given the type of environments we aim, health care systems, we added a specific type of filter to both approaches that allows us to specify a given instance of a topic. In the topic-based publisher/subscriber sub-layer, we may have topics such as *body temperature* or *blood pressure*; this filter permits to restrict a *body temperature* provided by a certain device. In the content-based publisher/subscriber sub-layer, we may have contents such as *body temperature* $> 30°$, as before, we may condition this event on a certain device.

B. Network Topology, Access Control and Data Forwarding

A system built with our middleware is composed of health devices and one or several monitoring equipment. The monitoring equipment can be a smartphone, a tablet or a notebook/computer. We call these equipment brokers, since they run the middleware.

A key role in the proposed system is the proxy server. This role is performed by one of the brokers. The proxy server has the responsibility of authenticating the access to the services and in forming the distribution table during the publishing of a service (data) in a broker. For each user, a profile is defined in the proxy server. This indicates which devices and service types a user may observe.

When a broker has a service to publish, it registers the service under his control in the proxy server. Every time another broker aims to subscribe this service, it sends the request type and user information for authentication to the proxy server. The proxy server verifies if the user has access, and if it is accepted, the proxy server notifies the broker responsible for the service, which registers the request.

In certain applications, we envisage scenarios where there is no common communication protocol between the broker that publish the data and the broker that subscribe it; e.g., a health device is connected to a smartphone without Wi-Fi but with Bluetooth connection and the subscriber has no Bluetooth. In this scenario, the broker may subscribe for the service through another broker. When a broker receives a subscription of a service, but he is not the proxy server (subscription forwarding), it forwards the request to the proxy server; upon receiving a valid authentication, it register the information in its forwarding table. The publisher is notified of the subscription by the proxy server. When the intermediate broker receives messages from the publisher it forwards them to the originating broker.

In the next section, we discuss the implementation of the middleware and of an application.

IV. Implemented Case Study

The middleware was implemented using the Java programming language. This choice is justified by some features of the language, namely, the object-oriented flavour, the strong-typed system and a thread model supported in the language. In the implementation of a layered software stack, an important trait is the definition of the API. We explore the interface construct in the java language that permits to define abstract object types decoupling the implementation of the lower layers from the upper one. The fact that the thread model is embedded in the language permits the implementation of a multithread solution without a strong coupling with the underlying operating system (OS). This allows reducing the development time of applications and its maintenance, as well as the deployment of the application in different operating systems. Also, the Java programming language is the base of the Android OS that is an important choice for a large market share in mobile devices, tablets and embedded boards. Another important fact is the freely availability of the Java virtual machine for any PC-based OS.

A. Study case: Blood Pressure and Electromyography Acquisition

As a study case, we implemented a distributed application to monitor the blood pressure, the acceleration and the electromyography signal in two subjects. We utilized the UA-767PBT [12] device by A&D for blood pressure monitoring for a subject. This device provides a Bluetooth interface. For the acceleration and the electromyography signal monitored in another subject, we used a device from shimmer [13] that permits to acquire several types of signals. The shimmer sensor is equipped with a Bluetooth interface and an 802.15.4 radio interface with a TCP/IP stack. We integrated each device in the middleware by implementing a virtual device adapter (section III-A) based on the documentation made available by the manufacturers. Two graphical interfaces were developed; one for the Android OS and another for a PC-based OS. In each interface (Windows OS or Android OS), the user can monitor the blood pressure value and choose which signal to visualize or the acceleration signal or the electromyography signal.

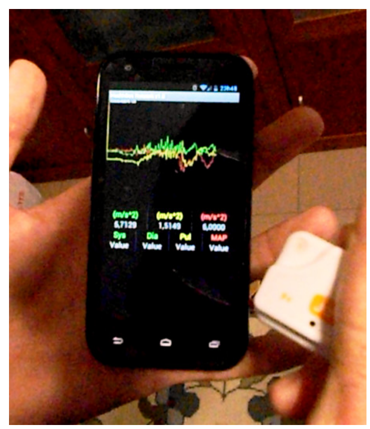

Fig. 2. Acceleration signal visualized in the smartphone.

In Fig. 3, we present the structure of our testing scenario. It consists of three elements: a smartphone, an embedded board and a notebook, each one with different capabilities in terms of clock frequency, memory and OS. The smartphone is a Huawei X3 IDEOS running Android OS version 2.3.3 with 256MB of memory and 600MHz of clock frequency. The embedded board [14] is based on the processor ARM Cortex-A8 from TI, which is a dual core processor with a clock frequency of 1.2 GHz, and 1 GB of memory, running Android OS, version 4.0. In this scenario, the shimmer device is connected to the smartphone using the Bluetooth protocol. The application in the smartphone depicts the acceleration and electromyography signals (Fig. 2 and Fig. 5) and publishes them to other hosts. The electromyography and the acceleration signals were acquired at 512 Hz. The smartphone is connected with the notebook using a Wi-Fi connection. The blood pressure device connects to the embedded board by Bluetooth. The embedded board publishes pressure values, which were subscribed by the notebook and the smartphone. The connection between the embedded board and notebook is by Bluetooth. Since it uses a Bluetooth connection, the blood pressure is published through the proxy server (notebook), which forwards it to the smartphone that subscribed this event. The proxy server also forwards the acceleration and the electromyograph signals to the embedded board. In Fig. 4, it is depicted an alarm (systolic value above a threshold) from the blood pressure device subscribed only by the notebook. In Fig. 5, we present simultaneously all devices. No delay was seen in the visualization of the signals in any of the elements (notebook, embedded board or smartphone).

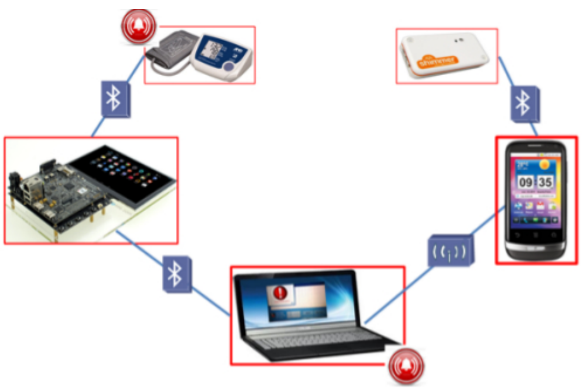

Fig. 3. Testing Scenario.

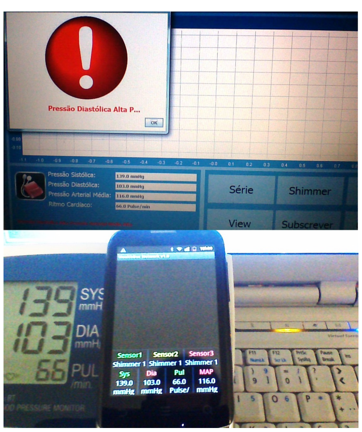

Fig. 4. Systolic blood pressure alarm.

Fig. 5. Blood pressure, acceleration and electromyography acquisition.

V. Conclusion

In this paper, we have presented a distributed middleware whose design aims to facilitate the integration of independent health devices, which use different communication standards. We have also described an application built with the middleware, which shows the devices seemingly integrated, allowing the remote monitoring as well as the reception of alarms. An advantage of this proposal is the integration of devices built by different manufacturers that use proprietary transaction protocols to extract information. Also the choice of the language permitted the deployment of the system in different operating systems (Windows, Linux, Mac OSX and Android).

As a future work, we aim to implement a routing strategy that takes the capability of the publisher into account, such that a broker with higher bandwidth can aggregate data and distributed it on behalf of a slower publisher.

Acknowledgment

The authors would like to thank Professor Graça Minas for kindly providing the shimmer device that was used in this work. This work has been supported by FCT Fundacao para a Ciencia e Tecnologia in the scope of the project: PEst-OE/EEI/UI0319/2014.

References

[1] M. J. OGrady *et al.*, “Towards evolutionary ambient assisted living systems,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 1, pp. 15–29, 2010.

[2] E. Jovanov *et al.*, “A wban system for ambulatory monitoring of physical activity and health status: applications and challenges,” in *Proc of IEEE Engineering in Medicine and Biology Society 27th Annual International Conference*. IEEE, 2006, pp. 3810–3813.

[3] C.-J. Su and C.-Y. Wu, “Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring,” *Applied Soft Computing*, vol. 11, no. 1, pp. 315–325, 2011.

[4] K. Lorincz *et al.*, “Sensor networks for emergency response: challenges and opportunities,” *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16–23, 2004.

[5] R. DeVaul *et al.*, “Mithril 2003: Applications and architecture,” in *Proc of 7th IEEE International Symposium on Wearable Computers*, 2003, pp. 4–11.

[6] N. Oliver and F. Flores-Mangas, “Healthgear: a real-time wearable system for monitoring and analyzing physiological signals,” in *International Workshop on Wearable and Implantable Body Sensor Networks*. IEEE, 2006, pp. 4–pp.

[7] U. Anliker *et al.*, “Amon: a wearable multiparameter medical monitoring and alert system,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 4, pp. 415–427, 2004.

[8] A. Wood *et al.*, “Context-aware wireless sensor networks for assisted living and residential monitoring,” *IEEE Network*, vol. 22, no. 4, pp. 26–33, 2008.

[9] K. Wac *et al.*, “Mobile patient monitoring: the mobihealth system,” in *Proc of IEEE Engineering in Medicine and Biology Society 31st Annual International Conference*. IEEE, 2009, pp. 1238–1241.

[10] P. T. Eugster *et al.*, “The many faces of publish/subscribe,” *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[11] R. Baldoni *et al.*, “Distributed event routing in publish/subscribe systems,” in *Middleware for Network Eccentric and Mobile Applications*. Springer, 2009, pp. 219–244.

[12] (April 2014). [Online]. Available: http://www.aandd.jp/products/medical/bluetooth/ua_767pbt.html

[13] (April 2014). [Online]. Available: http://www.shimmersensing.com/shop/shimmer-exg-module

[14] (April 2014). [Online]. Available: https://www.tigal.com/product/2543