mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications

Oresti Banos, Rafael Garcia, Juan A. Holgado-Terriza, Miguel Damas,
Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga

Research Center for Information and Communications Technologies of the University
of Granada (CITIC-UGR), C/Periodista Rafael Gomez Montero 2, Granada, Spain
`oresti@ugr.es`

Abstract. Mobile health is an emerging field which is attracting much attention. Nevertheless, tools for the development of mobile health applications are lacking. This work presents mHealthDroid, an open source Android implementation of a mHealth Framework designed to facilitate the rapid and easy development of biomedical apps. The framework is devised to leverage the potential of mobile devices like smartphones or tablets, wearable sensors and portable biomedical devices. The framework provides functionalities for resource and communication abstraction, biomedical data acquisition, health knowledge extraction, persistent data storage, adaptive visualization, system management and value-added services such as intelligent alerts, recommendations and guidelines.

Keywords: mHealth framework, mobile health, digital health, portable sensors, wearable sensors, biomedical sensors, health devices.

1 Introduction

The way healthcare services are delivered has radically changed during the last years. Recent surveys show a growing tendency in physician mobile health adoption. Mainstream medical applications are mostly devoted to learning and informative purposes [13]. Physicians increasingly recommend the use of health apps to patients [1]. While most apps require users to actively report about their health conditions, e.g., through annotating dietary habits [9] or daily routines [12], new technological trends seek to benefit from the information collected through wearable biomedical devices. For example, built-in smartphone sensors [10,14] or external wearable devices [11,8,6] may be used to detect abnormal conditions.

Mobile health (mHealth) is far from mature. Scientists still need to build and assess the complete spectrum of mHealth technologies. Powerful frameworks and tools that support the development and validation of multidisciplinary mHealth applications are required. Various attempts exist to this respect. For example, [7] provides an open source code for electrocardiogram signal processing. In [3] a mobile phone platform to collect users psychological, physiological, and activity information is presented. A mHealth middleware framework integrating

L. Pecchia et al. (Eds.): IWAAL 2014, LNCS 8868, pp. 91–98, 2014.

multiple interfaces and multiparameter monitoring of physiological measurement is proposed in [5]. Tools to analyze the provenance of mHealth data have also been suggested in [4]. These solutions focus on a specific domain or lack essential features for health applications. Therefore, this work proposes a novel mHealth Framework for the development of safe, scalable and effective applications.

2 Requirements of a mHealth Framework

The main goal of mHealth frameworks is to foster the research and development in health and medical domains as well as to accelerate the market of mobile health technologies and applications. The essential requirements needed in the design of a mHealth framework are outlined next. A certain level of abstraction from heterogeneous resources should be ensured to make hardware and its communication transparent to the developer. For the sake of interoperability, the framework should define a unified model for multimodal health data. Secure local and remote storage of health data is required to ensure persistence. The framework should provide techniques to extract health knowledge from raw medical and physiological data. Mechanisms to visualize medical and health information in a user-friendly fashion must be also provided for both average users and specialists. Another major requirement refers to the provision of healthcare services such as health delivery, personalized guidelines and intelligent recommendations. Finally, the framework should be modular and extensible to future sensor technologies and application needs.

3 Architecture of the mHealth Framework

In the light of the requirements defined in Section 2, a novel framework devised to enable the easy and agile development of mHealth applications leveraging on heterogeneous wearable biomedical devices is proposed. This mHealth Framework implements functionalities to support resource and communication abstraction, biomedical data acquisition, health knowledge extraction, persistent data storage, adaptive visualization, system management and value-added services.

Figure 1 shows the architecture that implement the functionalities and the components of the mHealth Framework. mHealth data delivered by mobile and biomedical sensors is collected and structured by the Communication Manager. This raw data can be stored in the Storage Manager, further processed by the Data Processing Manager, graphically represented by the Visualization Manager or directly used by the applications built on the mHeath Framework. Moreover, the medical knowledge derived by the Data Processing Manager can also be stored in the Storage Manager, input to advanced functionalities provided by the Service Enablers or used by the mHealth applications. Since the Storage Manager offers persistence, stored data can be offline processed by the Data Processing Manager, graphically represented by the Visualization Manager or accessed by the mHealth applications. Finally, the mHealth Framework offers, by means of the System Manager, functionalities to manage general resources of the mobile device.

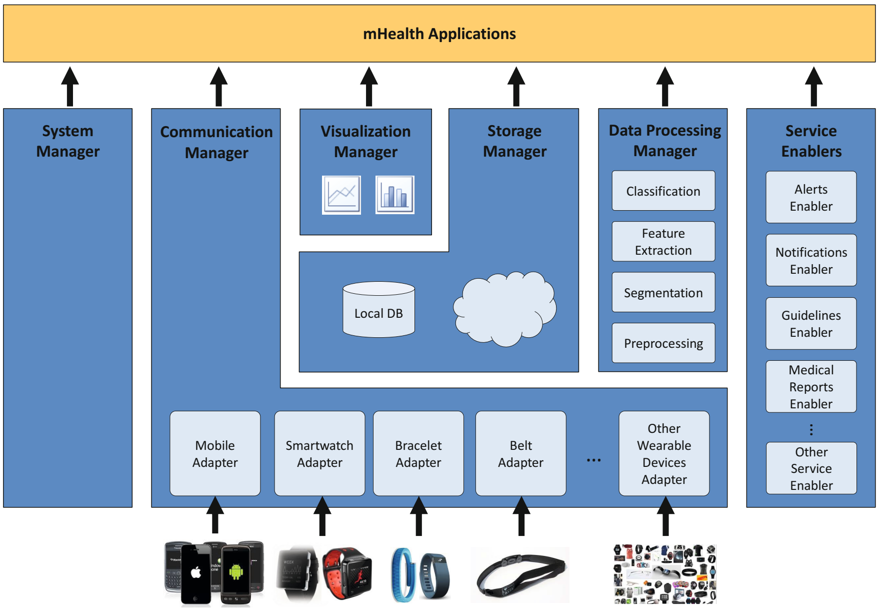

Fig. 1. mHealth Framework architecture

3.1 Communication Manager

mHealth applications may operate on multiple heterogeneous mobile and biomedical devices. The Communication Manager provides the abstraction level required to enable the functioning of applications independently of the underlying health technologies. This manager makes the communication transparent and serves as interpreter of the multimodal health data. In order to procure transparent communication and data retrieval, the Communication Manager incorporates Adapters, which are standalone modules devised to support the use of an specific mobile or biomedical device. The Adapter manages the connection with the device, interprets the received data and maps it to the unified data model (Section 3.7). The modularity of the Adapters makes the Communication Manager extensible and evolvable to future devices and technologies.

3.2 Storage Manager

The Storage Manager provides data persistence both locally and remotely. It enables the easy retrieval of stored data, abstracting the queries from the underlying storage system. This manager is also responsible for the efficient synchronization of the data and its secure transmission to the remote store, either in the cloud or remote server.

3.3 Data Processing Manager

This manager is in charge of the processing of health data by providing signal processing, data mining and machine learning techniques. These functionalities can run either on-the-fly by processing the data collected at runtime by the Communication Manager, or offline by retrieving the data from the Storage Manager. The Data Processing Manager includes four independent modules typically used in data processing.

Preprocessing. The collected health data may be affected by diverse type of artifacts such as spurious spikes or electronic noise, or be loosely controlled resulting in abnormal values and inconsistencies. Accordingly, it may be necessary to remove these anomalies from the raw data, e.g., by using filtering or screening techniques. This module is devised to apply mechanisms to clean, transform and ultimately adequate the data to the specific needs.

Segmentation. Biodata streams generally need to be split into segments or pieces. For example, sliding window approaches are commonly used for the partitioning of body-motion data. This module provides diverse techniques to split the data.

Feature Extraction. To provide a more tractable representation of the signals general or domain-specific features are extracted. Examples of features are statistical functions such as the mean or median, time/frequency transformations, heuristics, etc. This module permits to transform the input data into a reduced representation set of features or feature vector.

Classification. Artificial intelligence algorithms are widely used to gain knowledge from the collected health data. The features extracted by the Feature Extraction module are input to this type of algorithms provided by the Classification module to eventually categorize the data.

3.4 Visualization Manager

The data representation is a fundamental element of any mHealth app. Since applications may have different objectives and target users, developers require a wide sort of graphical representation tools. This manager is in charge of providing diverse modes and ways to display data. An online mode is identified for the depiction of the data provided by the Communication Manager, which corresponds to the information collected by the health sensors at runtime. On the other hand, an offline operation mode is defined for the visualization of data saved by the Storage Manager. Not only raw signals may be represented but also the information obtained after the data processing.

3.5 System Manager

The System Manager provides developers with functionalities to manage general resources of the mobile device. Examples of these resources are wireless connections (WiFi, 3G connection, Bluetooth), geopositioning technologies (GPS), screen configuration or battery management.

3.6 Service Enablers

An important characteristic of several mHealth applications is the intervention on health states. Health data may be profited to influence elements of the intervention and yield new information from which to act. This information is here devised to be provided to the users through a set of Service Enablers, which support advanced functionalities for alerts, notifications, guidelines and reports.

Alerts Enabler. This enabler provides mechanisms to trigger alerts and emergency procedures when abnormalities or risk situations are detected. Examples of these mechanisms are automatic phone calls and messages that may be delivered to the patients' family, carers and emergency services in the event of a critical situation (fall detection, cardiac event, etc.).

Notifications Enabler. Users may need to be timely or occasionally informed about important facts of their healthcare and wellbeing process. Health remainders (medication, workout, etc.) are essential mechanisms to engage users in the care process, to procure their organization and to empower them to meet the treatment goals. This enabler is devised to support prescheduled or event-based user-friendly notifications that may also trigger additional services.

Guidelines Enabler. Instructions, encouragements and educational information from specialists are of high value to promote healthy lifestyles and to support the patient self-care. This enabler provides multimedia tools for displaying guidelines that may be personalized and adapted to the user's needs and conditions.

Medical Reports Enabler. This enabler is devised to facilitate the structuring of the medical knowledge in an expert-oriented format. It may help clinicians and care professionals to interpret health trends and to support medical decisions.

3.7 Data Model

A unified Data Model enables the representation of heterogeneous health data and guarantees interoperability among the mHealth Framework components and applications. A generic, flexible and extensible model is of utmost importance due to the variety of available and future sensing modalities. The mHealth Data Model comprises five elements. The Session object is the main element and represents a recording session including its metadata. The Session is composed of several Sample objects which refer to each sample from the data stream collected during the session. Each Sample links to multiple Device objects which represent the devices streaming during the session. Since a device offers different sensor modalities, the Device links to the Sensor objects. The Sensor contains the data collected by a given sensor in a specific moment. Metadata is required to interpret the data collected by the multimodal sensors and the different devices. To reduce the model overhead and since the metadata does not vary during a session, the Metadata object is associated to the Session. The Metadata defines the types of sensors, the units of the measurements, the start and end time of the recorded session and the sampling rate.

4 mHealthDroid: mHealth Framework Implementation

mHealthDroid is the Android implementation of the proposed mHealth Framework, released open source[1] under the GNU General Public License version 3. mHealthDroid is devised to operate on the Android operating system 4.2 ("Jelly Bean"), although it provides backwards compatibility to version 2.3.3 ("Gingerbread").

The mHealthDroid Communication Manager has been implemented to facilitate the incorporation of new Adapters. To do so, it provides a generic Adapter skeleton. The current implementation of mHealthDroid provides the Adapter for Android mobile devices and the Adapter for the Shimmer3 wearable device [2]. The Android Mobile Adapter abstracts the sensors embedded into the mobile device, e.g., GPS, temperature or humidity. Likewise, the Shimmer Adapter provides the means to communicate the wearable device with the mobile device and map the data to the proprietary format. The Shimmer3 device provides multiple sensing modalities that span from inertial sensing via accelerometer, gyroscope, magnetometer, and altimeter, to physiological signs measurement such as electrocardiogram or electromyogram among others.

The Storage Manager incorporates a SQLite[2] database to implement the local persistence functionality. SQLite is a popular database engine on memory constrained systems, like mobile devices, since it runs in minimal stack space and very little heap. The Storage Manager also offers an interface to easily retrieve, based on diverse identifiers (session, device identifier, date, time interval), the data stored in the SQLite database. Database consistency check procedures are implemented by the Storage Manager to ensure integrity in the synchronization between the remote and local storage. The transmission to the remote storage is implemented using a HTTP POST request method which encloses in the request message's body the JSON[3] representation of the data. mHealthDroid also offers a server side implementation for remote persistence. This implementation builds on a MySQL[4] database and provides PHP scripts that use the MySQLi[5] API to manage the remote database.

The mHealthDroid Data Processing Manager provides an essential set of functionalities typically used in the data processing chain. The Preprocessing module implements two techniques: upsampling to increase the sampling rate and downsampling to reduce the sampling rate. A sliding window approach, widely-used in signal processing problems, is implemented by the Segmentation module. The Feature Extraction module implements some generic statistical features such as mean, variance, standard deviation, zero crossing rate, mean crossing rate, maximum and minimum. The Classification module builds on an open source stripped version of Weka[6]. It provides functionalities to train and validate

---

[1] Available at `https://github.com/mHealthDroid/mHealthDroid`

[2] `http://www.sqlite.org/`

[3] `http://json.org/`

[4] `http://www.mysql.com/`

[5] `http://www.php.net/manual/en/book.mysqli.php`

[6] `https://github.com/rjmarsan/Weka-for-Android`

machine learning models, that can be used for classification purposes. mHealthDroid currently implements Naive Bayes, Adaboost, Decision Trees, Linear Regression and ZeroR classification techniques.

The Visualization Manager builds on the open source library Graphview[7], which has been adapted to fulfill the particular needs of mHealth data representation. The manager allows multiplot visualization, multisignal representation and customization for diverse graph types.

The System Manager offers simple interfaces to access common mobile devices resources (WiFi, 3G, Bluetooth and screen) and builds on the standard Android API.

mHealthDroid implements three Service Enablers. The Alerts Enabler provides interfaces to trigger phone calls and text messages. The Notifications Enabler implements text remainders that can be scheduled in a simple way. Moreover, this enabler also provides advanced notifications that can trigger external functionalities or applications. Both Alerts and Notifications Enablers build on the standard Android API. Finally, the Guidelines Enabler provides interfaces to reproduce multimedia content, both locally and remotely stored. The Media Player Android[8] API is used in mHealthDroid to control playback of audio and video files for the local content. For the reproduction of remote multimedia content, the Guidelines Enabler implements a set of functions that build on the YouTube Android Player[9] API. This is particularly practical to access a huge variety of medical and wellbeing content.

An exemplary app[10] (see Figure 2) has been developed to illustrate the potential of the mHealthDroid implementation and to validate it.

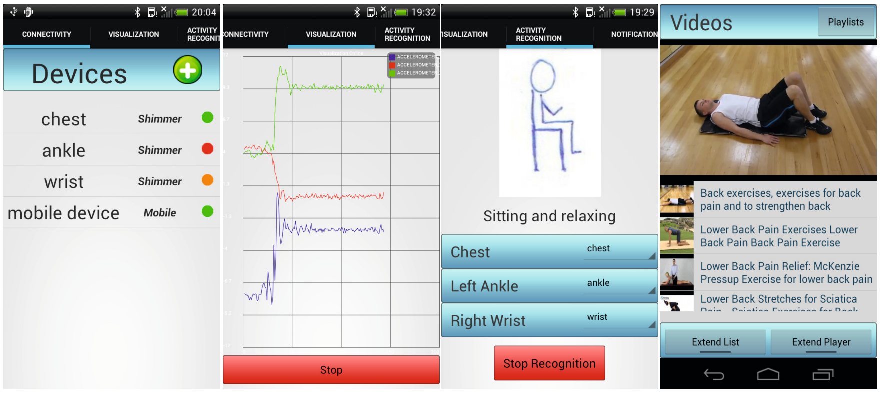

Fig. 2. Exemplary mHealth app developed using mHealthDroid

---

[7] `http://android-graphview.org/`

[8] `http://developer.android.com/reference/android/media/MediaPlayer.html`

[9] `https://developers.google.com/youtube/android/player/`

[10] Available in Google Play at `https://play.google.com/store/apps/details?id=com.mHealthDroid.activitydetector&hl=es_419`

5 Conclusions

This paper presents a novel framework devised to facilitate the development of mobile health applications in a simple and agile fashion. The framework has been designed taking into account the essential requirements of mHealth technologies and applications. Moreover, this work introduces mHealthDroid, an open source implementation of the proposed mHealth Framework that operates on the Android OS. The mHealthDroid initiative aims at bringing developers, healthcare professionals, academics and health enthusiasts together to exchange ideas and cooperate in the definition of valuable tools for a healthier world. Accordingly, the authors encourage the community to contribute to this innovative platform by supporting the use of the latest sensors, incorporating new behavioral algorithms or simply making use of it for the development of mobile health applications.

Acknowledgments. Work supported by the CICYT SAF2010-20558 and UGR Plan Propio PP2012-PI11 projects and the FPU Spanish grant AP2009-2244.

References

1. Taking the Pulse®(US). Technical report, Manhattan Research (2014)
2. Burns, A., et al.: SHIMMER. A Wireless Sensor Platform for Noninvasive Biomedical Research. IEEE Sensors Journal 10(9), 1527–1534 (2010)
3. Gaggioli, A., et al.: A Mobile Data Collection Platform for Mental Health Research. Pers. Ubi. Comp. 17(2), 241–251 (2013)
4. Prasad, A., et al.: Provenance framework for mHealth. In: 5th Int. Conf. on Comm. Syst. and Net, pp. 1–6 (2013)
5. Chen, P.H., Chen, H.M.: Framework design-integrating an android open platform with multi-interface biomedical modules for physiological measurement. Journal of Conv. Infor. Tech. 7(12), 310–319 (2012)
6. Banos, O., et al.: Physiodroid: combining wearable health sensors and mobile devices for a pervasive, continuous and personal monitoring. The Scientific World Journal (2014)
7. Oster, J., et al.: Open source Java-based ECG analysis software and Android app for Atrial Fibrillation screening. In: Comp. in Cardiology, pp. 731–734 (2013)
8. Oresko, J.J., et al.: A Wearable Smartphone-Based Platform for Real-Time Cardiovascular Disease Detection Via Electrocardiogram Processing. IEEE Trans. on Inf. Tech. in Biomed. 14(3), 734–740 (2010)
9. LoseIt®, `http://www.loseit.com/`
10. Habib, M.A., et al.: Smartphone-based solutions for fall detection and prevention: challenges and open issues. Sensors 14(4), 7181–7208 (2014)
11. Bsoul, M., et al.: Apnea MedAssist: Real-time Sleep Apnea Monitor Using Single-Lead ECG. IEEE Trans. on Inf. Tech. in Biomed. 15(3), 416–427 (2011)
12. MyFitnessPal®, `http://www.myfitnesspal.com/`
13. Khalifian, S., et al.: Medical student appraisal: searching on smartphones. Appl. Clinical Inf. 4(1), 53–60 (2012)
14. Mazilu, S., et al.: Online detection of freezing of gait with smartphones and machine learning techniques. In: 6th Int. Conf. on Perv. Comp. Tech. for Healthcare, pp. 123–130 (2012)