Seamless Wearable Data Collection in a Mobile Environment

Fabio A. Seixas-Lopes [1], Carlos Lopes[1], Maria Marques [1] and Carlos Agostinho[1]

[1] *Centre of Technology and Systems, UNINOVA, 2829-516 Caparica, Portugal*

Abstract

As the world progresses, technology developments provide innovative approaches to contemporary circumstances. The interest in wearables for health and wellbeing has increased and enabled the growth of digital health solutions. Since data is the foundation for developing eHealth systems and its capabilities, a seamlessly integration of data sources enhances the interoperability between the existing heterogeneous solutions, serving as basis for services that allow monitoring and managing habits that can greatly impact health and wellbeing. Citizen Hub is a mobile application, designed to seamlessly integrate data from wearables and medical devices. Its objective is to allow connecting different devices in an aggregated way and provide feedback and insightful information about wellbeing and health. The collection can be configured, through a straight-forward and clean interface, and uploaded in a structured way to the cloud platforms, enabling additional features and services. Citizen Hub approach thrives to deliver a complete and dynamic solution for ICT interoperability, while maintaining security, trustworthiness and privacy methodologies as well as compatibility with relevant standards. In this paper we explore the development, functionalities and processes (such as BLE Communication and Reverse-Engineering) for the integration of devices.

Keywords

eHealth, Digital Health, Smart Devices, IoT, Data Collection, Data Integration, Interoperability.

1. Introduction

Health systems in Europe have continuously recognized and integrated digital health solutions to improve care delivery, increase efficiency, conveniency and reducing costs [1]. With the increase demand for portable and easy-to-access solutions, together with the development of handheld phones, wearables and other devices, the Internet of Things paradigm gains shape as a mean for developing meaningful, interoperable and personalized health care digital solutions and patient-centric innovation [2]. The easiness of these applications extends the engagement and benefits beyond the reach of the medical care facilities [3] [4].

Data stands as the key component for processing the world and deliver services upon the process of understanding that data, so it is essential to have a proficient and easy-to-access solution for data collecting [5]. The promotion of standards and data harmonization is an effective methodology for heterogeneity and compatibility with existing solutions or areas where there already exists significant progress, thus enabling further interoperability [6]. It is also important to implement adequate data policies, regarding the personal data processing, consent and other aspects from GDPR [7], to ensure citizen protection.

Smart4Health [8] is a European project that aims to tackle these issues with the creation of a citizen-centered platform for the management of electronic health records. With data as the foundation for the project, the integration of data sources stands as a way to enhance the interoperability between existing heterogenous solutions. Since 19% of the European Union population (age 16-74) uses smart watches or other smart wearables (and the percentage has the potential to grow even more) [9], these devices are an existing method and accessible way for citizens to collect wellbeing and health data.

Proceedings Name, Month XX–XX, YYYY, City, Country
EMAIL: email1@mail.com (A. 1); email2@mail.com (A. 2); email3@mail.com (A. 3)
ORCID: XXXX-XXXX-XXXX-XXXX (A. 1); XXXX-XXXX-XXXX-XXXX (A. 2); XXXX-XXXX-XXXX-XXXX (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

Citizen Hub emerges as a mobile solution for Smart4Health, acting as gateway for the user to access the Citizen Health Data Platform and upload information using only one application, a Hub. The objective of Citizen Hub is mainly to provide a clean and straight-forward user interface with seamless data collection capabilities that allow for the citizen to effortlessly connect devices, such as wearables, and integrate that data to take advantage of the Smar4Health benefits.

This paper explores the applicability and development of Citizen Hub as a digital health solution for the collection of data in wearables in a mobile environment. In the Device Integration section, the standards, protocols and processes (such as Bluetooth Low Energy (BLE) Communication and Reverse-Engineering) needed to integrate the wearables and other devices are reviewed and briefly explained. The following section dives into the implementation process of the previously explained concepts, which are tested and validated in the ensuing chapter. Finally, some conclusions and future work are presented.

2. Device Integration

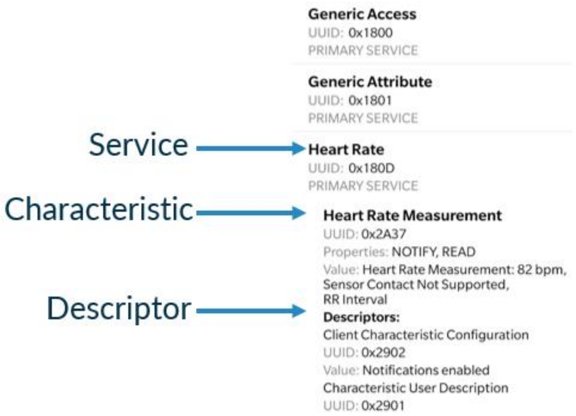
Wearables, smart devices and sensors can provide useful information for its user. Aggregating this information means collecting, filtering and analyzing, processing it so it can be usefully displayed to the user or passed to other services, connecting real-time data streams from the rapidly expanding set of IoT sensors and networks [10]. The gathering and transformation of this information allows to paint a clearer picture of a certain person's wellbeing, activity and present a complementary tool for health assessment and diagnosis. Typically, to benefit from multiple data sources, a user needs to install different applications where data is presented and stored on the manufacturers' terms. Citizen Hub contribution to this heterogenous and impractical environment is the benefit of using only one application, a Hub, that collects, stores and displays data using only the citizen's preferences, giving the ownership of the data back to the user.

The BLE protocol [11] is Citizen Hub most used communication protocol. It is briefly described below as well as the process for integrating external data sources.

BLE is a widely adopted standard, part of the Bluetooth 4.0 core specification, and presents a viable solution for an optimized version of the classic Bluetooth in terms of efficient energy consumption [12]. It presents a specific architecture that specifies layers from the host, controller and applications. For this paper, we will focus on two layers related to the host, that show how information is exchanged, the Generic Access Profile (GAP) and the Generic Attribute Profile (GATT). The GAP provides a framework that defines how BLE devices interact with each other, defining how parameters and data for advertising, scanning and connecting is defined. The GATT defines the data exposed by a BLE device. The most important data provided are the characteristics and services attributes.

A **service** is one or more attributes that specify a general functionality of a device. A **characteristic** is part of a service and specifies a particular information that the device can provide. An example of three attributes can be seen in **Figure 1**, where the general heart rate service is specified as a primary service with a standard 16-bit **UUID** (Universally Unique Identifier) "0x180D" [13], so other devices can read that information knowing it represents a heart rate. The characteristic is a heart rate measurement, which also as a standard UUID, which is related to the service and specifies that the value provided by this characteristic is a single measurement of heart rate with additional information about the reading (specific context flags representing the sample). The descriptor acts a state check, showing if notifications or indications are enabled.

To interact with a device there are some possible operations such as a write command (which writes without need for confirmation), requests (for read and write values), responses (answers the requests), notifications (so the server sends information when the characteristic value has changed) and indications (same as notification but needs acknowledgment). With these means of interaction, BLE provides an optimized way for devices to communicate by keeping the transportation to a minimum and sending small amounts of data at low speeds.



Generic Access
UUID: 0x1800
PRIMARY SERVICE

Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Heart Rate
UUID: 0x180D
PRIMARY SERVICE

Heart Rate Measurement
UUID: 0x2A37
Properties: NOTIFY, READ
Value: Heart Rate Measurement: 82 bpm, Sensor Contact Not Supported, RR Interval
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications enabled
Characteristic User Description
UUID: 0x2901

Figure 1: Example of attributes for a BLE device.

For integrating external data sources, it is important to highlight the need for UUIDs and standards. As seen in the example presented in **Figure 1**, the heart rate service was compliant with the Bluetooth standard for a heart rate service. If this was not the case, and since devices output more than one type of data at the same time, with multiple services and characteristics, at potentially high speeds, it can be very time consuming to try to understand what values correspond to what measures, or even understand what the device really measures at all. If compliant, the behaviour of devices is predictable, we know what the information means and we can develop a generic algorithm for our communication handler for all heart rate services of all devices. Considering the current heterogeneity of contemporary wearables and since the manufacturers tend to prefer that the users use the manufacturers application to take advantage of keeping the measured data, only a small percentage of devices have compliance with standards, which presents a challenging scenario to attempt to construct a versatile wearables hub.

The process of integrating external data sources can be split into different sequential steps:

1. *Identify Device Capabilities/Features*. When identifying viable devices for integration, it is necessary to evaluate their capabilities with a considerable amount of testing to understand how they perform and the accuracy of their data. It is also important to understand if the type of data provided is of use for the intended application.
2. *Identify Supported Communication Protocols and Standards Compliance.* For a device to be implemented, it must have a compatible supported communication protocol. If it follows communication standards, the behaviour is predictable and the integration process is faster. If, for example, UUIDs match features of the device, we can infer which characteristics or services corresponds to each feature.
3. *Extract Information about Functioning.* This process is commonly the basis for reverse-engineering [14], which is the methodology to discover and defining an object's functioning by various methods such as specific testing and comparation. The objective is to understand how the device communicates to allow Citizen Hub to interact with it, by decoding the patterns and isolate certain device behaviours. In this step, other methodologies such as packet sniffing (a technique whereby packet data flowing across the network is detected and observed) are used to see what messages and information flow across the device-phone connection.
4. *Model communication messages.* After understanding the device functioning, the next step is to model communication messages to enable the data collection and data transport. Here, we establish the client-server relationship. This may include processes such as connection, authentication, acknowledgement and other communication messages or operations (write, read, notify, indicate, etc.) needed for the device to be useful.
5. *Create Connection Handler*. The handler is represented by an agent, which is part of Citizen Hub and is responsible for handling the connection and communication with the device, ensuring the data reaches the application and the device behaves as intended.
6. *Process, Analyze and Display Data.* After the data enters Citizen Hub, it is processed and analyzed, to be later displayed to the user in a useful manner and consumed by other services or functionalities.

3. Implementation

Considering the information presented in the previous section, it is possible to model a methodology for device integration considering each step provided. If the compliance for a certain device is there, we can rapidly move through the initial steps and have the communication messages modelled and start to work on a connection handler, that is able to control that device or any other device that has that service or characteristic. If not, the first steps will likely include some ad-hoc intermediate operations to handle the uncertainty of having, for example, random UUIDs for characteristics. Despite the different difficulty and time consumption of each scenario, the next step is to create a communication handler, that we simply call an **agent**. To accomplish this, the Citizen Hub application uses its main service to create and manage multiple instances of these agents, each one regarding different devices. These agents contain a device UUID, the device features (heart rate, steps, posture, etc.), protocols to trigger, write and read to the various services and characteristics of the device, listeners to receive incoming notify/indicate messages and additional data algorithms to filter values from the messages and to tag additional data such as timestamps. The agent is created when a new device is connected and runs continuously while the device provides information, establishing a bridge of data between the device and the hub. This relation is depicted in ***Figure 2***, a sequence diagram for a typical Bluetooth interaction between a device and Citizen Hub.

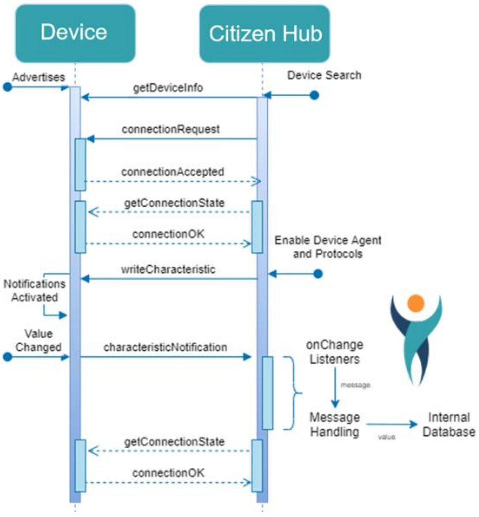
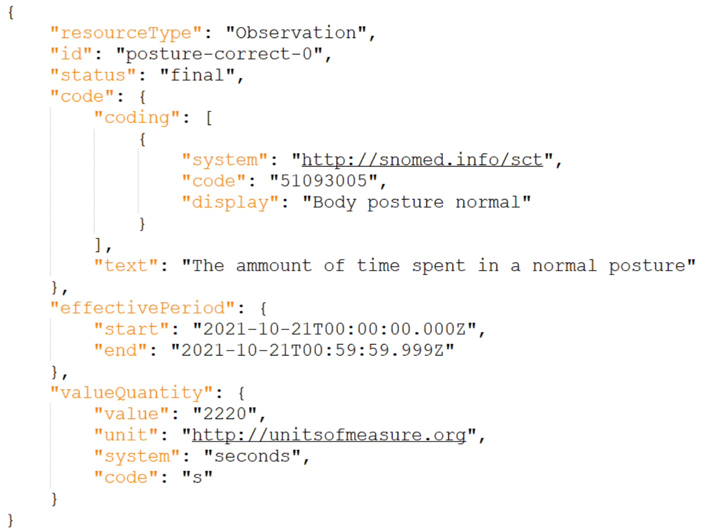


Figure 2: Typical Bluetooth interaction in Citizen Hub.

After the connection is established, Citizen Hub creates an agent for the device which writes to a characteristic it wants to listen to. When new information arises, the device notifies Citizen Hub, which handles the message/values and stores them internally. The device and Citizen Hub keep the connection alive until it is closed (user preference) or disconnects (due to distance, Bluetooth being turned off, etc.).

To allow data re-use and easy sharing with several platforms and systems, the data structure is compliant to FHIR [15], a common standard for health care data exchange that aims to structure and simplify interoperability, and an example observation is depicted in **Figure *3***.



```
{
    "resourceType": "Observation",
    "id": "posture-correct-0",
    "status": "final",
    "code": {
        "coding": [
            {
                "system": "http://snomed.info/sct",
                "code": "51093005",
                "display": "Body posture normal"
            }
        ],
        "text": "The ammount of time spent in a normal posture"
    },
    "effectivePeriod": {
        "start": "2021-10-21T00:00:00.000Z",
        "end": "2021-10-21T00:59:59.999Z"
    },
    "valueQuantity": {
        "value": "2220",
        "unit": "http://unitsofmeasure.org",
        "system": "seconds",
        "code": "s"
    }
}
```

Figure 3: Simplified posture observation in FHIR.

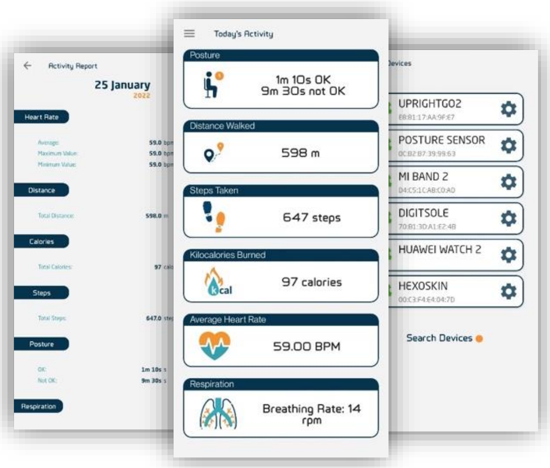
In this example, a simplified posture observation is presented, where the *code* fields mention the standard for the observation type (which is described in the *text* field), the *effectivePeriod* field specifies the time window on which the observation has occurred and the *valueQuantity* describes the observation values and units (which also follow a standard).

4. Test and Validation

From the technical point of view, some challenges on the implementation process had to be overcome. The lack of standards in low price Bluetooth devices made extremely difficult to directly associate services or characteristics to features. Several reverse-engineering processes were conducted to correctly integrate the devices. The inconsistency of BLE devices, that communicate at short ranges and low speeds, required the implementation of many reconnecting methods to keep the devices synchronized. Also, it was necessary to develop mechanisms to prevent the mobile operating system to terminate any tasks or services from the application while doing its battery and process optimization, which varies for each mobile manufacturer. To identify and tackle these issues several tests were performed to validate the approach, such as connecting different devices with different configurations and test their limitations, force them to communicate with other applications to understand their *modus operandi* and testing the overall limits of the BLE protocol.

Currently, Citizen Hub gathers information efficiently from a set of fully integrated devices, including a smart insole, a smart vest, posture sensors, fitness bands, blood pressure monitors, smartwatches and a gateway to provide information on low back posture training/physiotherapy (developed in the context of the Smart4Health project). For these, the information presented is consistent with the devices and within valid intervals for each type of data and the connections are stable, within the BLE context.

From usability point of view, a simple and intuitive interface (**Figure *4***) was developed. A helper supports the user with some information during any process and the application is multilingual. The interface includes a section for the current day activity, daily activity reports from previous days, a devices section that allows for searching, configure and update devices, a section for managing the user account and an about page with additional information. There is no complex data details or analytics and only basic direct information from the devices and their connection state are presented.



Activity Report
25 January
2022
Heart Rate
Average: 59.0 bp
Maximum Value: 59.0 bp
Minimum Value: 59.0 bp
Distance
Total Distance: 598.0 m
Calories
Total Calories: 97 calo
Steps
Total Steps: 647.0 ste
Posture
OK: 1m 10s s
Not OK: 9m 30s s
Respiration

Today's Activity
Posture: 1m 10s OK
9m 30s not OK
Distance Walked: 598 m
Steps Taken: 647 steps
Kilocalories Burned: 97 calories
Average Heart Rate: 59.00 BPM
Respiration: Breathing Rate: 14 rpm

Devices
UPRIGHTGO2 E8:81:17:AA:9F:E7
POSTURE SENSOR 0C:B2:B7:39:99:63
MI BAND 2 D4:C5:1C:AB:C0:AD
DIGITSOLE 70:B1:3D:A1:E2:4B
HUAWEI WATCH 2
HEXOSKIN 00:C3:F4:E4:04:7D
Search Devices

Figure 4: Screenshots from the Citizen Hub.

In addition to this, and demonstrating the cross-platform functionally, Citizen Hub is also being used in the **Smart Bear** project [16], which has successfully integrated the Citizen Hub application in their cloud (exchanging data and functionalities). This demonstrates the potential of this interoperable solution to integrate any platform that is receptive or compliant with standardized data.

5. Conclusion and Future Work

The Citizen Hub application is currently published, free of charge and available to any citizen registered on the Smart4Health platform. The functionalities described are stable, and within the planned specifications and standards. Future work includes the integration of additional devices and systems. Also, the release of new versions and firmware for already implemented devices, will require continuous updates. With an increasing amount of data available and with the integration of new devices and new types of data, new and more complex functionalities are also planned.

6. Acknowledgements

This work has been developed in the context of Smart4Health project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826117.

7. References

[1] U. Deetjen, "European E-Prescriptions : Benefits and Success Factors," no. 5, pp. 1–21, 2016.

[2] D. Kalra, Z. Kolitsi, and C. Schulz, "DHE Consultation with citizens/patients on the European Health Data Space," no. June, 2020.

[3] D. Lewkowicz, T. Slosarek, S. Wernicke, A. Winne, A. M. Wohlbrandt, and E. Bottinger, "Digital therapeutic care and decision support interventions for people with Low Back Pain: Systematic review," *JMIR Rehabil. Assist. Technol.*, vol. 8, no. 4, pp. 1–18, 2021.

[4] C. Imison, S. Castle-clarke, R. Watson, and N. Edwards, "Delivering the benefits of digital health care," 2016.

[5] J. Hentschel, "Contextuality and data collection methods: A framework and application to health service utilisation," *J. Dev. Stud.*, vol. 35, no. 4, pp. 64–94, Apr. 1999.

[6] EU4Digital, "Common Guidelines for eHealth Harmonisation and Interoperability," no. December, 2020.

[7] E. Union, "EU General Data Protection Regulation," *Off. J. Eur. Union*, 2016.

[8] "Smart4Health." [Online]. Available: www.smart4health.eu.

[9] Eurostat, “Community survey on ICT usage in households and by individuals,” 2020.

[10] S. Tang, D. Shelden, C. Eastman, P. Pishdad-Bozorgi, and X. Gao, “A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends,” *Autom. Constr.*, vol. 101, no. May 2019, pp. 127–139, 2019.

[11] Bluetooth SIG, “BLE Definition,” 2021. [Online]. Available: https://www.bluetooth.com/.

[12] K. Townsend, C. Cufí, Akiba, and R. Davidson, *Getting Started with Bluetooth Low Energy*. O’Reilly Media, Inc, 2014.

[13] Bluetooth SIG, “16-bit UUID Numbers Bluetooth Document,” 2022.

[14] H. A. Miiller, J. H. Jahnke, D. B. Smith, M. A. Storey, S. R. Tilley, and K. Wong, “Reverse engineering: A roadmap,” *Proc. Conf. Futur. Softw. Eng. ICSE 2000*, pp. 47–60, 2000.

[15] H.-H. L. S. International, “Fast Healthcare Interoperability Resources,” 2021. [Online]. Available: https://www.hl7.org/fhir/.

[16] “Smart Bear Project.” [Online]. Available: https://www.smart-bear.eu/.