Evaluation of a Software Architecture Supporting Android Applications for Users with Motor Disabilities

Olibário José Machado-Neto
University of Sao Paulo
olibario@icmc.usp.br

Maria da Graça C. Pimentel
University of Sao Paulo
mgp@icmc.usp.br

ABSTRACT

We propose a software architecture to help developers to create Android-based applications for users with motor disabilities – specifically. It supports using one or more hardware components of mobile devices including built-in sensors, camera and microphone, and encompasses an easy way of using and integrating such resources, which may lead to applications that provide alternative ways for accessing and managing data by users with disabilities. The solution also provides functionalities to work with raw sensor data, and offers a model for storing medical information of users. An evaluation with 19 software developers indicate that the architecture can be useful for creating not only solutions for people with motor disabilities, but diverse applications.

KEYWORDS

Mobile Computing, Accessibility, Assistive Technology

1 INTRODUCTION

We conducted an extensive literature review, through which we noticed that the literature lacks a software architecture to facilitate the development of applications for people with disabilities by using and integrating the hardware of mobile devices. Actually, the use of software architecture to support accessibility is well documented [6] [1] [7]. However, none of the studies we found focus on the creation of applications for mobile devices. Other works explicitly present accessible solutions by using resources of mobile devices, but software architecture is not their focus[2][3][4][5].

In this study, we present an architecture to fulfill these needs. The differences between our architecture and the ones we found in the literature are: it is intended for the mobile context; it can be used to record medical data; it integrates the built-in hardware of mobile devices in a straightforward way in order to create applications that be used for assistive purposes.

2 METHODOLOGY

We revealed the main requirements of our architecture after carrying out three two-hour sessions of brainstorming were carried with two occupational therapists. They are: 1) the assistive solutions will consist of applications that explore the resources of mobile devices; 2) the integration of the resources must be simple to implement; 3) the applications must be easy to modify because user requirements may change over time; 4) data processing will be carried out primarily in the mobile device; 5) data storage will be be carried out primarily in a cloud-based server.

We performed controlled activities in which developers could implement Android applications using our architecture. We offered a free 20-hour Android programming course for developers and designers who had prior experience with Java programming language. The course was held from Monday to Friday, four hours a day. Nineteen participants concluded our course.

We presented our software architecture in the last day of the course and guided the students in the creation of a mobile application for taking pictures. The action of opening the camera should be performed by both clicking a button and shaking the device. Also, the device should vibrate while opening the native Android camera application. An additional screen of the application should show the values of the accelerometer, the gyroscope and the light sensor, simultaneously.

At the end of the course, the developers were invited to answer a questionnaire regarding the software architecture, with the following questions: Does the architecture help to: 1) create interfaces of applications that use the resources of mobile devices? 2) create applications that uses the camera? 3) create applications that use the microphone? 4) create applications that use: a) the accelerometer? b) the gyroscope? c) the magnetometer? d) the proximity sensor? e) the light sensor? f) the humidity sensor? g) the thermometer? h) pressure sensor? 5) create applications that integrate more than one hardware resource? 6) create applications for people: a) with motor disabilities? b) with other disabilities?

All questions were answered according to a Likert scale from 1 (does not help) to 5 (helps a lot).

3 RESULTING SOFTWARE ARCHITECTURE

Our architecture follows a client-server structure, in which the device is responsible for processing data and the server is in charge of storing data (Figure 1).

The interaction user interface of the Mobile Client is represented by the component “Visualization.” From the interface, the application accesses the control layer, which contains pre-programmed XML files with examples of interfaces for accessing the basic data of the camera, the microphone and the sensors.

The component “Sensors” contains Facade classes implemented in Java with the main functionalities of each sensor of Android devices. The functionalities provide the developer with the possibility to retrieve the raw hardware data, or processed data.

After the information is processed by the application, the data can be stored in the “Data” layer. In order to adequate the users’ information to Health parameters that might be useful for the specialists, the layer “Data” is equipped with JSON-schemas, which

In: Sessao de Pôsteres do WebMedia’2017, Gramado, Brasil. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres. Porto Alegre: Sociedade Brasileira de Computação, 2017.

ISBN 978-85-7669-380-2.

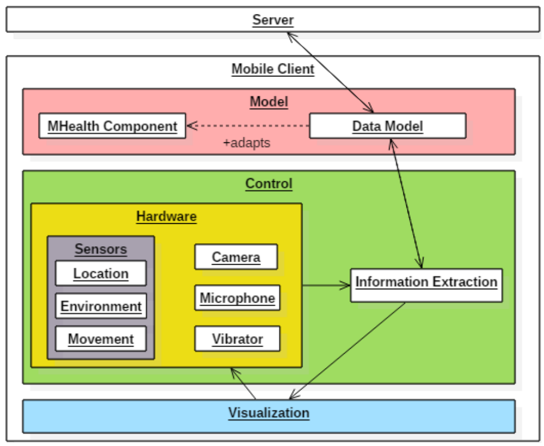

Figure 1: Software architecture Evaluated

define the correct structure and attributes for medical parameters that are relevant in Health applications[1], by using a repository with recommendations of how to store medical data.

4 EVALUATION

The ages of the 19 participants varied from 20 to 56. Of them, 16 were under 30. Sixteen were male and 3 were female. The answers to our questionnaire are summarized in Table 1.

Table 1: Summary of the responses. The columns show the number of responses in each of the Likert scales (1 to 5).

Response related to	1	2	3	4	5
interfaces of applications	0	0	3	8	8
microphone	0	1	3	9	6
camera	0	1	3	8	7
accelerometer	0	0	1	8	9
gyroscope	0	0	1	11	7
magnetometer	0	1	4	11	3
pressure sensor	0	0	5	12	2
proximity sensor	0	0	2	11	5
light sensor	0	0	2	10	7
humidity sensor	0	1	5	11	2
thermometer	0	0	8	8	3
sensors integrated	0	0	1	9	8
motor disabilities	0	0	0	8	11
other disabilities	0	0	1	5	13
TOTAL	0	4	39	129	91

The 19 participants of the course included one developer with more than 2 years of experience in Android programming and 18 developers with less than 6 months of experience. One developer with less experience answered that the architecture "helps a little" to use the humidity or the magnetometer. Another one, also with little experience, answered that the architecture "helps a little" to create applications that use the microphone or the camera. The magnetometer, humidity sensor, pressure sensor and thermometer were the ones with the largest number of neutral responses: 4, 5, 5 and 8, respectively. This was expected since the application did not require the use of these sensors. However, this might also indicate that the architecture must simplify even more the access to these resources. The experienced participant answered 3 (neutral opinion) regarding the help of the architecture for the sensors: humidity, magnetometer, thermometer, proximity and pressure. For all other questions, his answers were 4 ("it helps").

A total 79% of the participants replied that the architecture either helps or helps a lot to create applications that use the camera or the microphone. All other answers were from 3 to 5 in the Likert scale. Two participants answered 5 (helps a lot) to all of the questions.

The sensors that received the most negative responses were the thermometer and the humidity sensor, which were not available in any of the devices used. However, even in these cases the percentage of responses with 4 or 5 values were 57.9% for the thermometer and 68.42% for the humidity sensor. All other sensors received at least 73,68% of good acceptation by the participants.

[1]Available at http://www.openmhealth.org/documentation/#/schema-docs/schema-library

5 CONCLUSION

We presented a software architecture along with an evaluation with 19 software developers. The results indicate that architecture is useful for creating applications that demand using one or more resources of mobile devices. The evaluation also allowed identifying components that demand improvement.

The results of the guided tests with software developers infer that the architecture may be useful for developing not only applications for accessibility purposes that use and store medical data, but also for general-purpose applications. This seems to be specially true for inexperienced developers.

ACKNOWLEDGMENTS

The authors would like to thank FAPESP, Capes and CNPq for the financial support of our research.

REFERENCES

[1] R. Baguma and J. T. Lubega. 2008. A Web Design Framework for Improved Accessibility for People with Disabilities (WDFAD). In *Proc. 2008 Intl. Cross-disciplinary Conf. on Web Accessibility (W4A) (W4A '08)*. 134–140. https://doi.org/10.1145/1368044.1368077

[2] Z. Chen, M. Lin, F. Chen, N.D. Lane, G. Cardone, R. Wang, T. Li, Y. Chen, T. Choudhury, and A.T. Campbell. 2013. Unobtrusive sleep monitoring using smartphones. In *2013 7th Intl. Conf. on Pervasive Computing Technologies for Healthcare and Workshops*. 145–152.

[3] J. Feijó Filho, T. Valle, and W. Prata. 2012. Breath Mobile: a Software-Based Hands-Free and Voice-Free Breathing Controlled Mobile Phone Interface. In *Proc. 14th Intl. ACM Conf. on Computers and Accessibility*. 217–218.

[4] N. Ohnishi, T. Matsumoto, H. Kudo, and Y. Takeuchi. 2013. A System Helping Blind People to Get Character Information in Their Surrounding Environment. In *Proc. of the 15th Intl. ACM Conf on Computers and Accessibility (ASSETS '13)*. ACM, Article 34, 2 pages. https://doi.org/10.1145/2513383.2513389

[5] D. Oliveira, M. Carvalho, and T. Paixão. 2016. Ubiquitous Computing: Gestures Interaction Applied the Learning Disabilities in Process Literacy. In *Proc. of the 22nd Brazilian Symposium on Multimedia and the Web (Webmedia '16)*. ACM, 107–110. https://doi.org/10.1145/2976796.2988168

[6] K. Upatkoon, W. Wang, and S. Jamin. 2005. WebBee: An Architecture for Web Accessibility for Mobile Devices. In *Proc. IFIP Intl. Conf. on Personal Wireless Communications*.

[7] F. Zacarias, M. Balderas, R. Cuapa, and A. Tellez. 2009. Mobile Architecture for Question Answering. In *Proc. of the 7th Intl. Conf. on Advances in Mobile Computing and Multimedia (MoMM '09)*. ACM, 569–573. https://doi.org/10.1145/1821748.1821858