Quantitative and Qualitative Evaluation of a Navigation Application Adapted to Young People with Intellectual Disabilities

Halisdemir Ufuk
Capgemini Engineering R&I department
Illkirch-Graffenstaden, France
e-mail: ufuk.halisdemir@capgemini.com

Besnier Morgane
Capgemini Engineering R&I department
Illkirch-Graffenstaden, France
e-mail: morgane.besnier@capgemini.com

Zegarra Flores Jesus
Capgemini Engineering R&I department
Illkirch-Graffenstaden, France
e-mail: jesus.zegarraflores@capgemini.com

Coutant Rémi
Centre Ressources Enfance & Adolescence
ADAPEI du Territoire de Belfort, Belfort, France
e-mail: r.coutant@adapei90.fr

***Abstract*— The "ADAPEI Transport" app is an urban navigation tool for young people (10 to 20 years old) with intellectual disabilities. These disabilities lead to memory, time, and space perception problems. Using this tool, these young people learn how to use public transport and navigate in the city independently thanks to a reference path created by an educator. Along the path, steps are recorded to help the user interact with his/her environment in the navigation part. Currently, we are in the test and evaluation phase of the project for the navigation part. The aim of this article is to show the evaluation methods we have created (Python tool and questionnaire) to have feedback from real tests and to test the efficiency of the app. In fact, the app collects data from Inertial Measurement Unit (IMU) sensors and Global Positioning System (GPS), allowing to replay the journey done in a real test. The first results using the app in a real situation with a disabled youth and an educator are very encouraging. Additionally, thanks to the questionnaire and the Python tools developed to qualitatively evaluate a journey from user input, we were able to improve the app.**

Keywords - intellectual disability; user interface; mobile app; algorithms.

I. INTRODUCTION

The association ADAPEI Belfort [1], an association of parents, friends and people with intellectual disabilities or affected by mental disorders, and Capgemini Engineering [2] have developed an inclusive application for young adults and people having intellectual disabilities. This collaboration led to the creation of a mobile application which is now used within the association as a tool to help people having intellectual disabilities to take public transport and to navigate independently in urban areas. For these young people, it is often difficult to use standard navigation applications (Google maps, City mapper) because the interface is not adapted, often too complex or with inaccurate information. In addition, they are more likely to lose their phone, which means that they often have low-cost phones, with less accurate sensors and/or poor Internet reception, which can make using these applications even more difficult. The ADAPEI Transport application is in an advanced stage of development and is now in the test and validation part. This app allows people having intellectual disabilities to gain autonomy to walk independently in cities using adapted navigation information and security buttons [3].

The focus of this work is to create methodologies to test the efficiency of the interaction between the interface of the app and people having intellectual disabilities using quantitative and qualitative evaluations. In the state of the art, we have found some articles that evaluate health mobile apps from a qualitative point of view. For example, there is a method called Mobile Apps Rating Scale (MARS) that evaluates the quality of applications according to four components: engagement, functionality, aesthetics, and information quality [4]. However, this method is not necessarily adapted to the evaluation of the app with people having intellectual disabilities for increasing autonomy. Another article explains the use of a tool to analyze the interaction between a navigation app and people with visual impairment [5]. There are different cameras to record the movement and displacement of the person. In our case, adding these cameras could be disturbing and intrusive for the intellectually disabled person.

In this article, we first briefly describe the ADAPEI Transport app. We then present the evaluation method of the application set up for the test and validation phase. In this part, we will present the creation of a questionnaire, the creation of the tool in Python to evaluate the journey in a qualitative way, and the comparison of navigation algorithms to test their efficiencies. Then, we will present the results and, finally, the conclusion.

II. THE ADAPEI TRANSPORT APP

The application is composed of three parts: (i) the creation of a route done by a caregiver and/or young adult with disabilities, (ii) the visualization of this route to learn the steps and (iii) the navigation part. The first two parts have been tested with encouraging results. In this article, we will focus on the evaluation of the navigation part. The navigation part consists of showing a list of steps in a sequential way, depending on the position of the person. A progress bar is also shown to know if the person is close to getting off the bus or

 ISBN: 978-1-68558-004-9

to pressing the stop button. We have implemented two types of navigation algorithms to test their efficiencies: navigation by the nearest position of a step and navigation by circles of steps.

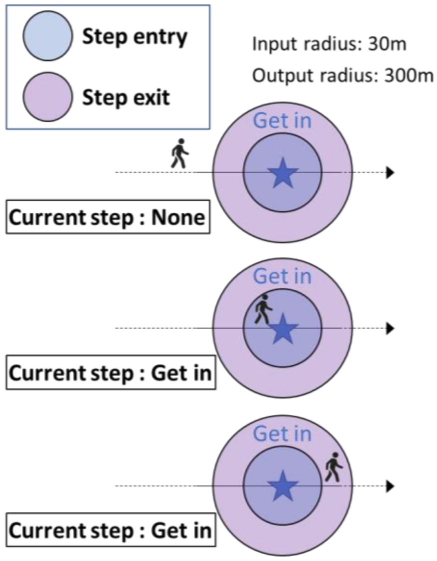

Figure 1. Navigation by circles

During the navigation, a file in JSON format is created containing the navigation data with the coordinates (latitude, longitude, GPS precision and speed) and the values of different sensors (compass, gyroscope, accelerometer). Using this file as input, we developed a tool (coded in Python) to replay the journey, to know when information is launched on the screen and to determine the behavior the user adopts.

III. METHODOLOGY

A. Questionnaire

To make a qualitative evaluation, a questionnaire was created. This questionnaire is filled by a caregiver who will follow the child/young adult during real-life tests. It has three objectives: to note the problems that the user may have during the journey (discomfort, direction problems, hesitations), to note the problems related to the application (bugs, display errors) and to propose improvements.

The questionnaire is inspired by various articles talking about metrics used in the qualitative evaluation of mobile apps [6][7]. We have chosen different metrics, such as 'efficiency, understanding, satisfaction'. The questionnaire is divided into 3 parts:

- Profile of the young person: age, transport habits, level of autonomy.
- Test process: errors in the journey, hesitation, intervention of the companion.
- The application: bugs encountered, remarks, inconsistency of information.

B. Python tool

To perform functional tests of the mobile application, a Java Android emulator in Android Studio was created replaying a trip using a GPS eXchange Format (GPX) file. This file is created when the user launches a trip on their phone. However, the Android emulator does not consider important elements such as speed, GPS accuracy and a set of data from the IMU sensors. This is the main reason why a more sophisticated emulator using Python has been developed. The purpose of this interface is to replay the trips in order to compare the navigation algorithms and to find possible problems/bugs and/or improvements.

The tool allows to recover many files: one to have the initial base of the journey, another to detect the changes of directions and segments to distinguish walking paths from bus paths, and the JSON file to replay the route done by the child using sensors data.

C. Algorithm comparison and slowdown detection

1) Navigation by the nearest position of steps

To know which step is the current step, the algorithm computes the distances between the current position of the person to all the positions of every steps. The step with the smallest distance is the current step and previous steps disappear.

2) Navigation by circles of steps

This algorithm aims to be as close as possible to reality. It has two circles per step which differentiate the entry from the exit of the step. Each circle radius depends on the type of the step. In this version, we detect if we are in the circle or not (see Figure 1). When we exit a step, all the previous steps disappear.

Example: *To enter the step "Get into the bus" you must be less than 30m away from the position of the step and to validate it, you must be inside the bus and more than 300m away from the position of the step and at a speed ~70km/h (bus speed).*

3) Slowing down detection

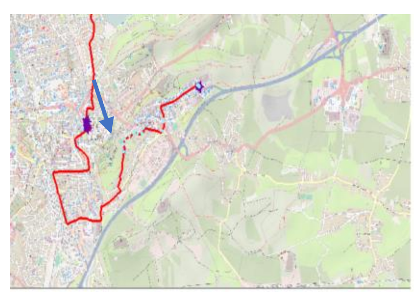

Figure 2. Belfort route, navigation by the nearest position of steps

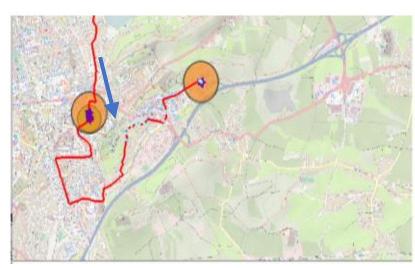

Figure 3. Belfort route, navigation by circles of steps

 ISBN: 978-1-68558-004-9

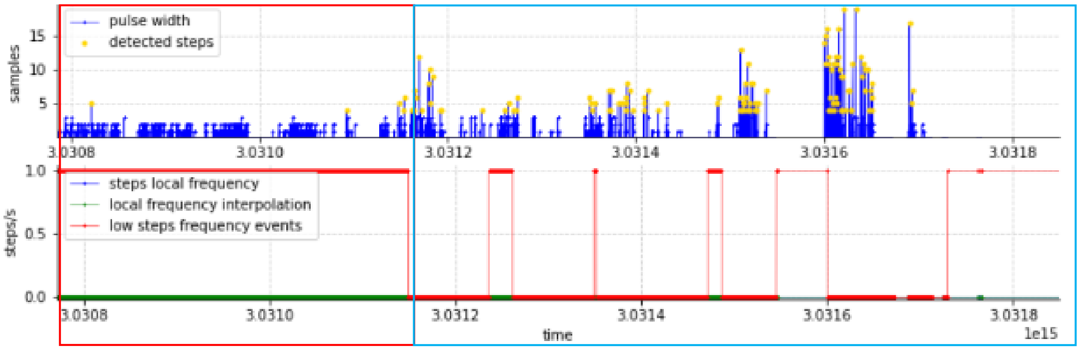

Figure 4. Slowdown detection graph

When a caregiver creates a journey using the app, a file that allows to split the trip into different segments and to differentiate the pedestrian segments from the bus/tramway segment is recorded. Using this file and the accelerometer data, we set up a slowdown detection in order to tell if the user could have some hesitation during the trip. The slowdown is detected using the frequency of the walking. A low step frequency can be considered as an hesitation of the person.

To identify if the user is walking at a low steps frequency, the compound acceleration is calculated thanks to the accelerometers data on three axes (X, Y, Z) (1).

$$a_{comp} = a_x + a_{xy} + a_z \quad (1)$$

The mean steps frequency for each segment is used to know if the user is walking at a low frequency (2). When the local step frequency is lower than 0.5*mean steps frequency, a low frequency event is raised

$$freq_{local}[i] = \frac{1}{time_{step_{i+1}} - time_{step_i}} \quad (2)$$

This algorithm was developed by our team and has an accuracy of 90% for the step counter [8].

IV. Results

A. *Algorithm's comparison*

In Figures 2 and 3, we see that, for the same position, the remaining steps are not the same. The algorithm by the nearest step takes the step with the smallest distance. At this point, the closest step is n°7. Therefore, steps 5 (to press on the stop button to alert to get off) and 6 (get off from the bus) are already validated, even though they have not been done. This can mislead the user or confuse them, as they do not see what the steps already validated consist of. Following this observation on this route and three other routes, we conclude that the navigation algorithm by circles seems to better reflect reality because steps 5 and 6 appear/disappear at the right moment when the person is close to getting off.

B. *Test in real conditions carried out in Belfort*

The objective of this part is to see the errors in real tests using the Python tool and the questionnaire as well as the slowing down of the user which we assume to be user hesitation.

A trajectory was created in Belfort (see Figure 2) with some characteristics: 13 km; a bus change, two moments to press on the stop button. Before being able to do the tests in real conditions, there is a whole learning process that the specialist does with the young person. Currently, we have only had one test feedback giving interesting results. The test was carried out with a teenager having intellectual disabilities who is not used to take public transport, but with a good level of mobility. He seemed comfortable during the test, but had a problem which forced the educator to act. The problem came from a confusion if it was the moment to press on the stop button or not. The progress bar was not completely full when the stop button should have been pressed. This problem led to changes in the application. Overall, this test was a success because, apart from this problem, everything else went well. In the questionnaire, it was also suggested to make a modification to have a bigger size of the step image to have a clear understanding of the current step to follow.

Thanks to the interface, we were able to visualize the problem with the progress bar that was wrongly implemented in the application while it was working in the Python tool. The tool made it easier to understand the error.

Figure 4 shows the slowdown. The segment in red represents the second bus trip and the blue corresponds to the last pedestrian part. We can see that the person does not walk on segment 3 as he is in the bus, so we do not have any step detected. On the other hand, we see slowdowns on segment 4 of variable duration. However, these slowdowns are interpreted as hesitations because the questionnaire does not tell anything about them. This can be due to a wait at a red light, for example.

V. Conclusion

In this paper, we developed methodologies to qualitatively and quantitatively test the efficiency of a navigation application designed for people with disabilities. For the qualitative evaluation, we developed a questionnaire is filled by a caregiver who follows the child/young adult during real-life tests. This allows us to get feedback as close as possible from the user's perspective and modify the app when necessary. For the quantitative evaluation, we developed

Python tools that allow to virtually replay a journey with different navigation algorithms to test their efficiency.

The methodology proposed seems to be correct. Thanks to the Python tool, we were able not only to identify problems present in the application, but also to compare the algorithms used for the navigation to see which one was the most efficient thanks to tests carried out on 4 routes.

Even if we only have one feedback on the questionnaire, it gave us very interesting information that allowed us to improve the application. More tests are still necessary to validate our methodology.

In the future, we should improve the slowdown detection using data from the other IMU sensors present on the phone to try to detect real hesitation.

REFERENCES

[1] Capgemini Engineering, France, [online] https://capgemini-engineering.com/fr/fr/, [last access October 2022].

[2] Adapei – Belfort territory, France, [online] https://adapei90.fr/, [last access October 2022].

[3] J. Zegarra Flores et al., “ADAPEI-TRANSPORT: A GPS Based Mobile App for Learning Paths and Improving Autonomy for Young Adults Having Intellectual Disabilities to Take Public Transport”, Computers Helping People with Special Needs. ICCHP 2020. Lecture Notes in Computer, pp. 112-119, 2020.

[4] Y. Terhorst et al., “Validation of the mobile application rating scale (MARS).” PLoS One, 15(11), e0241480, 2020.

[5] I. Maly, J. Balata, O. Krejcir, E. Fuzessery, and Z. Mikovec, “Qualitative measures for evaluation of navigation applications for visually impaired.” In 2015 6th IEEE international conference on cognitive infocommunications (CogInfoCom) , pp. 223-228, 2015.

[6] N. N. Patel and P. Dalal, “Usability evaluation of mobile applications”, International Journal of Engineering Research & Technology (IJERT), 2(11), pp. 299-302, 2013.

[7] R. Alturki and V. Gay, “Usability attributes for mobile applications: a systematic review.” Recent Trends and Advances in Wireless and IoT-Enabled Networks, pp. 53-62, 2019.

[8] J. Z. Flores, S. Cornus, E. Charbonnier, and L. Rasseneur, “Navi Campus: Quantitative Methodology for Evaluating the User Interface of a Navigation App Using Eye Tracker and Smartphone”, in the Proceedings of The Sixth International Conference on Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing (HEALTHINFO), IARIA, 2021, pp. 37-40.