Supporting Localization Testing through Automated Application Navigation

Lais Pereira Felipe[1,2], **Breno Miranda**[1]

[1] Informatics Center – Federal University of Pernambuco (UFPE)
Recife – PE – Brazil

[2]Post Graduate Program in Computer Science
Course level: Master's Degree, Start date: Feb-22, End date: Dec-24

{lpf2,bafm}@cin.ufpe.br

Abstract. *Software testing is necessary to ensure the delivery of a functional product characterized by high quality and reliability. This requires effort to develop a product with minimal issues. To ensure a positive user experience across diverse cultures, countries, and languages, the execution of Localization (l10n) and Internationalization (i18n) tests becomes imperative. Nevertheless, there is not much support of automated tools for this type of testing. This research aims to propose and evaluate automated solutions for supporting internationalization and localization testing.*

Resumo. *Testes de software são necessários para garantir a entrega de um produto funcional caracterizado por alta qualidade e confiabilidade. Isso requer esforço para desenvolver um produto com um número mínimo de falhas. Para garantir uma experiência de usuário positiva em diversas culturas, países e idiomas, a execução dos testes de Localização (l10n) e Internacionalização (i18n) torna-se imperativa. No entanto, não há muito suporte de ferramentas automatizadas para esse tipo de teste. Esta pesquisa tem como objetivo propor e avaliar soluções automatizadas para apoiar testes de Internacionalização (i18n) e Localização (l10n).*

1. Research Problem

Software targeting the global market, with its diverse users across the world, must ensure a consistently positive user experience for individuals from various linguistic and cultural backgrounds. To achieve this, software companies need to embrace software globalization (G11n) practices, adapting the content presented within the application to cater to each region, culture, and country it supports.

The G11n process comprises several aspects, including internationalization (i18n) and localization (l10n). Internationalization entails developing a software product in a way that enables it to seamlessly accommodate multiple languages and regions [Xia et al. 2013]. Conversely, localization guarantees that the product is finely tuned to meet the linguistic and cultural nuances of the local audience [Zhao et al. 2010].

L10n and i18n testing play a crucial role in ensuring that software products are appropriately adapted to cater to the diverse needs of global audiences. This includes not only validating textual content but also considering other essential aspects, such as the

orientation of objects on the screen. Even though it's an important process to provide an optimal user experience, there is a lack of automated tool support for l10n testing [Ramler and Hoschek 2017].

In this context, the anticipated outcome of this is the creation of a tool that could help testers identify changes in text across various languages. Additionally, it would accurately highlight problems related to language and localization issues, which could help testers and linguists cut down significantly on the time needed compared to manual checks and also increase coverage.

This paper is organized as follows: Section 2 presents basic concepts and gives an introduction to the subject. Section 3 presents the context of this research. Section 4 describes the methodology adopted and the evaluation of the tool developed. Then Section 5 defines the preliminary results reached. Section 6 presents related work, and section 7 declares the expected results.

2. Background

Localization and Internationalization tests are an essential component of the software development process, particularly when it comes to catering to a diverse global audience. These tests are crucial in ensuring that the software is capable of functioning seamlessly and effectively across a wide range of locales and that the language and cultural nuances of the users are appropriately accommodated. The primary purpose of i18n and l10n testing is to verify the accuracy and appropriateness of textual content, known as strings, in various languages and cultures.

The localization process ensures that contents like language, date, time, and currency adhere to the language requirements and cultural differences to ensure that the software product can be used effectively and comfortably by users worldwide. Also, it's important to make sure that expressions/slangs are still in use in the locale. Thus providing the users with a good experience while using the app.

However, i18n/l10n testing is not limited to textual content alone. Other crucial aspects, such as the orientation of objects on the screen, must also be taken into consideration. A prime example of this is the difference in reading direction between regions that read from right to left (RTL) and those that read from left to right (LTR).

The differences between the elements orientations impact the eye's movement on screens. For people who are used to reading contents from LTR, when looking at an app screen, they will naturally look first to the left corner of the screen. On the other hand, a user whose own first language is written in RTL will instinctively first look at the top right corner of the screen and then move their eyes to the bottom left corner of the screen [Abufardeh and Magel 2009].

To further elaborate, l10n/i18n testing must ensure that the graphical elements of the user interface (UI) are appropriately positioned. For example, in RTL regions, the placement of elements such as the back arrow would be on the right side of the screen, while in LTR regions, it would be on the left side.

Successful localization requires more than just accurately translating textual content. Other critical elements, such as navigation buttons, symbols, and icons, must also be appropriately positioned to accommodate the reading directions of different languages

and cultures. Neglecting these aspects can lead to confusion and frustration among users, negatively impacting the overall user experience.

3. Context and Motivation

Performing localization testing is critical to ensure the quality of a globally available software product. [Leiva and Alabau 2014] debate the importance of visual contextualization in user interface (UI) localization. They argue that language translation alone is not sufficient for effective localization, and that the visual context of the UI elements must also be taken into account.

Validation activities can include looking for newly added or modified strings and checking if they follow the locale's (linguistic and regional) specifications. Additionally, the tester should also identify if the content is completely shown, if the elements are positioned correctly, and if the content displayed does not convey a negative or aversive meaning to the user.

It is true that languages are in constant change, and the string values should meet the requirements. Moreover, layout improvements are continuous and consequently impact the way the strings are shown. Hence, every time a string value is modified or a new string is added, it is mandatory to validate the way it is exhibited. However, most of the time, the tester doesn't know exactly where to find the string.

The process of conducting i18n/l10n testing in applications can be a complex and time-consuming task for testers. They must constantly navigate through the app interface to locate and verify the accuracy of newly added or modified content. Unfortunately, in most cases, there is no accompanying information indicating the precise location of the screen where the string is presented. As a result, the process of identifying and validating the correct string in its intended context becomes a manual exploration activity, which can be particularly challenging for testers with limited experience on the apps. The more extensive the app, the more challenging it becomes to locate strings.

There are some tools available for the exploration and mapping of application strings, but those tools don't usually identify and highlight l10n/i18n issues. In addition, there are tools like the one mentioned in [Abufardeh and Magel 2009], which is primarily tailored to address RTL related issues exclusively.

Therefore, the development of a tool that could effectively not only map out a way to find these strings but also already point out possible issues and report them back would significantly contribute to the i18n/l10n testing process. Such a tool would not only improve the efficiency of string localization but also reduce the burden on testers.

For instance, let's examine Figure 1, which depicts screens of an Android app. The leftmost figure represents the main screen, while the rightmost figure represents the final screen. To navigate through the app, users are required to follow a sequential path from one screen to the next. The string "+ More Options" was recently added and needs to be validated in Spanish. Beforehand, the tool had already explored the application's contents in English and mapped out its interface. As a result, the tool can provide a sequence of screens that lead to the target string, helping the tester locate the desired string efficiently when evaluating the Spanish version of the app. Additionally, the tool's prior exploration may have already highlighted any potential issues during the exploration.

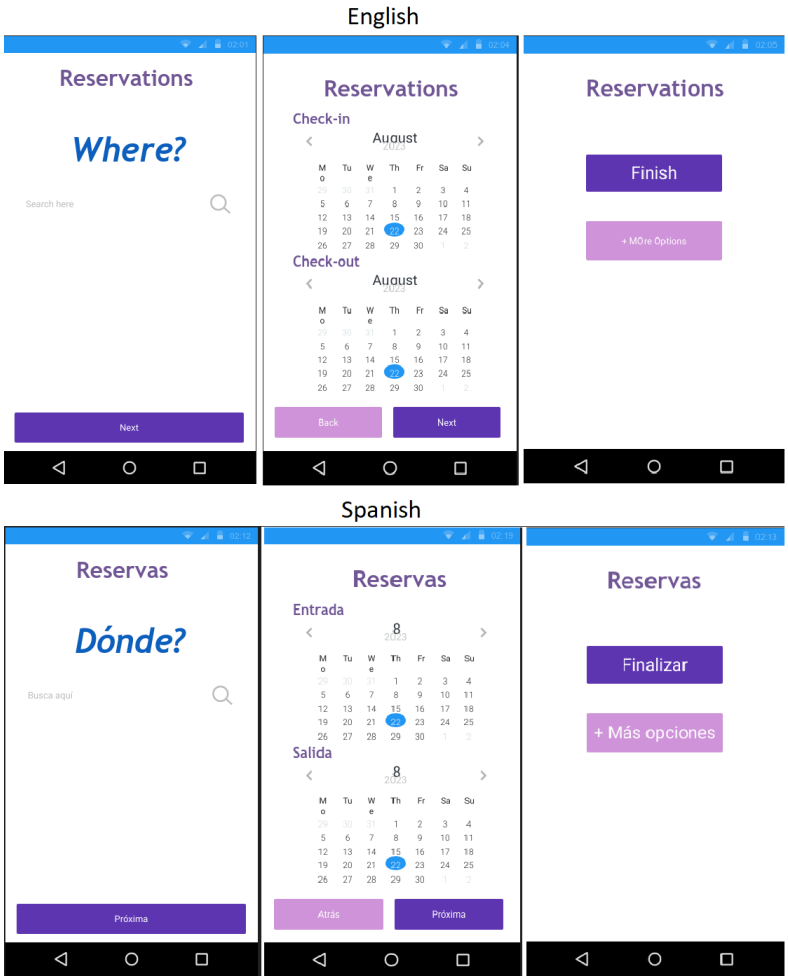

Figure 1. Localization Testing in English.

Based on the stated objectives, three research questions were identified: RQ1) *Does the tool actually contribute to the i18n/l10n testing process by making it faster and reducing manual effort?*; RQ2) *Is the tool assertive when pointing out i18n/l10n related issues?*; RQ3) *What coverage of i18/l10n related issues does the tool provide?*

4. Methodology

The methodological procedures were defined according to the vision of [Marconi and Lakatos 2003]. As for the nature, this is an applied research because it aims to generate knowledge that can be applied in the solution of practical problems or in the improvement of existing processes and technologies [Marconi and Lakatos 2003]. Regarding the objectives, it is experimental because it aims to establish relationships of cause and effect between variables through controlled manipulations of experimental conditions [Marconi and Lakatos 2003]. This research will assess the impact of the implementation and application of the tool in the test environment, and therefore it is experimental. About the research approach, this research will be qualitative and quantitative since feedback will be collected from users and also analyzed data such as time and failure coverage. Relating to the technical procedures, two approaches will be adopted. The first will be observation, which is the third phase of the research described

below and the beginning of the fourth phase, where data such as observed time and coverage percentage will be collected. The second is the questionnaire, which will be applied with a view to getting feedback from the user with respect to the tool. The development of this work will occur in phases as shown in Figure 2.

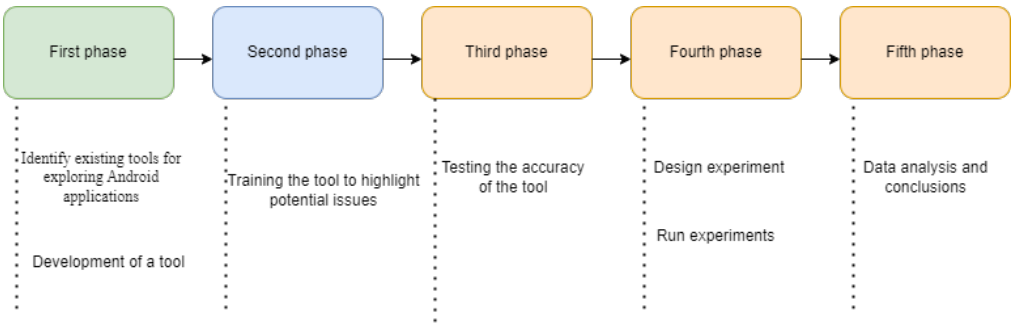

Figure 2. Research phases.

4.1. Development of a tool to support localization and internationalization testing

During the first phase, we will conduct research to identify existing tools for exploring Android applications. Simultaneously, we will create a tool that can consolidate the outcomes of this exploration and provide the mapping for the strings contained in the application. Moving into the second phase, our objective is to train the tool to effectively highlight potential issues detected during the exploration process.

4.2. Evaluation

In the third phase, our focus will be on testing the tool's ability to accurately identify i18n/l10n issues. To achieve this, we will provide Android applications with known i18n/l10n issues in order to help measure the tool's precision and coverage. The fourth phase evaluates the reduction of effort and time, where users will carry out manual searches for strings using the tool, and the time will be counted in each scenario (manual or automated with the tool), and a questionnaire will also be provided to the users about using the tool. In the fifth phase, the data obtained will be analyzed, and it will be observed whether the objectives were achieved and the questions answered. If the objectives are achieved, the tool can be applied to real i18n/l10n test scenarios since during the experimentation it was validated.

5. Preliminary Results

As of this submission, we are currently in the midst of our second phase. The initial development of a tool designed to map out strings within applications has been successfully completed. Our current focus lies in the identification of potential i18n and l10n issues associated with the strings that have been captured during the exploration process.

6. Related work

[Abufardeh and Magel 2009] presents an automated solution for testing the localization of Android applications, with a focus on RTL languages. Through the use of optical character recognition and specific linguistic rules, the approach verifies the correct implementation of localization in RTL languages. The work also presents a tool to assist i18n/l10n

testing, but it is focused on RTL testing only. [Ramler and Hoschek 2017] highlights the importance of i18n and l10n testing within software product development. It provides insights into the challenges, strategies, and tools that can enhance the internationalization and localization efforts of software companies. By embracing comprehensive testing methodologies and leveraging specialized tools, software developers can create products that effectively reach a global audience while minimizing post-development adjustments. Couto and Miranda developed L10N-TRAINER [Couto and Miranda 2023a, Couto and Miranda 2023b], a tool to assist in the training of novice i18n/l10n testers. The tool seeds faults in the string files from mobile open source applications in order to reproduce the failures found in i18n/l10n testing. The seeded applications can be used in the training allowing the novice testers to have the opportunity to learn about the practice and get in touch with i18n/l10n failures during their training stage.

7. Expected Results

As part of this research, the envisioned outcome is the development of a tool that empowers testers to swiftly locate altered or novel string values within diverse languages. This tool is anticipated to drastically reduce the time required compared to manual investigation. Additionally, there is an expectation that the tool will accurately pinpoint issues associated with i18n and l10n, effectively identifying instances of localization failures. The aspiration is for this tool to offer substantial coverage of such failure scenarios. Such a tool could be used in the industry to assist i18n/l10n testers.

References

Abufardeh, S. and Magel, K. (2009). Software internationalization: testing methods for bidirectional software. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 226–231. IEEE.

Couto, M. and Miranda, B. (2023a). An industrial experience report on the challenges in training localization and internationalization testers. In *Proceedings of the 8th Brazilian Symposium on Systematic and Automated Software Testing*, SAST '23, New York, NY, USA. Association for Computing Machinery.

Couto, M. and Miranda, B. (2023b). l10n-trainer: a tool to assist in the training of localization (l10n) and internationalization (i18n) testers. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, SBES '23, New York, NY, USA. Association for Computing Machinery.

Leiva, L. and Alabau, V. (2014). The impact of visual contextualization on ui localization.

Marconi, M. d. A. and Lakatos, E. M. (2003). *Fundamentos de Metodologia Científica*. Atlas, São Paulo, Brazil.

Ramler, R. and Hoschek, R. (2017). Process and tool support for internacionalization and localization testing. *Product-Focused SW Process Improvement*, pages 385–393.

Xia, X., Lo, D., Zhu, F., Wang, X., and Zhou, B. (2013). Software internationalization and localization: An industrial experience. pages 222–231.

Zhao, C., He, Z., and Zeng, W. (2010). Study on international software localization testing. *2010 Second World Congress on Software Engineering*, 2:257–260.