Towards Graphical User Interface Redefinition without Source Code Access: System Design and Evaluation

Rita Maia
Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
rita_carolina_maia@iscte-iul.pt

J. C. Silva
Escola Superior de Tecnologia, 2Ai, Instituto Politécnico do Cávado e do Ave, Barcelos, Portugal
jcsilva@ipca.pt

José Luís Silva
ITI/LARSyS, Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR-IUL Lisbon, Portugal
jose.l.silva@m-ti.org

Abstract
Nowadays several interactive computing systems (ICSs) still have Graphical User Interfaces (GUIs) that are inadequate in terms of usability and user experience. Numerous improvements were made in the development of better GUIs however, little has been done to improve existing ones. This might be explained by the fact that most ICSs do not provide source code access. In most cases, this means that only persons with source code access can (easily) enhance the respective GUI.

This paper presents a tool using computer vision (CV) algorithms to semi-automatically redefine existing GUIs without accessing their source code. The evaluation of a new GUI obtained from the redefinition of an existing GUI using the tool is described. Results show statistically significant improvements in usability (reduction of interaction mistakes), improved task completion success rate and improved user satisfaction.

Author Keywords
Engineering Interactive Computing Systems; Graphical User Interface; Redefinition; Usability.

ACM Classification Keywords

H.5.2 User Interfaces; H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous; D.2.2 Design Tools & Techniques.

Introduction

Graphical User Interfaces (GUIs) were subject to significant improvements along the years. Those progresses are the results of the conjunction of several advancements such as, better development approaches (e.g. user-centered) and evaluation techniques (e.g. empirical user testing supported by eye tracking [1]), new and improved widgets and interaction methods. The area of interface and interaction design made also important advances developing principles, guidelines, heuristics, standards [2–4] and even patterns for effective interaction design [5]. Those progresses are responsible for the arise of GUIs providing better usability and User Experience (UX). Besides all this work many existing interactive computing systems (ICS) still have inadequate GUIs. This might be explained by two reasons: (i) developers do not follow or do not correctly apply most recent advancements; (ii) old ICSs, and no updated version was developed. This work focuses on the latter case i.e., providing a solution to improve existing GUIs without having access to the source code of the respective ICS. One can argue that old ICS will tend to disappear or be updated. This might be true for some of them, but the fact is that several others will remain unchanged. Nowadays, many organizations are still using old ICS with inappropriate GUIs e.g., several US government IT systems (see US Government Accountability Office report: https://www.gao.gov/assets/680/677454.pdf, last accessed: May 23, 2019).

Many ICS were developed without properly taking in consideration, usability and UX concerns. This paper addresses this challenge by presenting a tool, GUIRT (Graphical User Interface Redefinition Tool) enabling semi-automatic GUI redefinition without source code access and aiming to improve usability and UX. The benefits of the tool are illustrated with an example where an old GUI is redefined and evaluated.

Related Work

The work of Dixon et al. enables the identification (via computer vision algorithms) of a set of GUIs widgets without source code access [6]. This identification supports pixel-enhancement of the GUI but not its redefinition [6, 7]. The ISI tool [8] enables the integration of independent ICSs into only one GUI adapted to the specificities of the users. This work focuses on facilitating user interaction and improving efficiency using automation but does not support GUI redefinition. Gaganpreet et al. [9] developed a solution to improve GUIs taking into consideration user's psycho-physiological state however, the GUI redefinition is done manually and case by case. Silva et al. [10] described an ongoing work describing an initial set of design guidelines to change GUI at runtime based on both user's specificities and emotional state. Other works enable automatic GUI generation adapted to user's features but not the redefinition of existing ones [11-13].

System Overview

The developed tool is based on existing CV algorithms for GUI widget extraction without source code access and Sikuli [14] scripts to automate the execution of the task in the original GUI. Sikuli is an automation tool that uses image recognition to identify and interact with

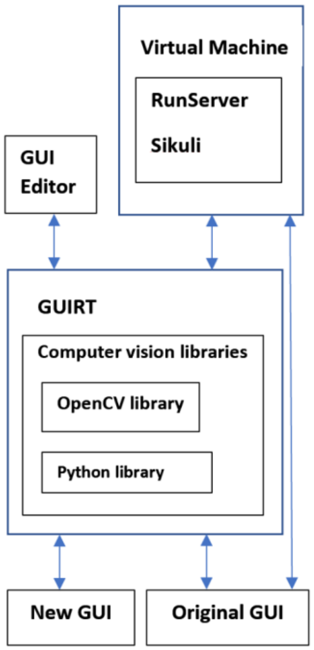

Figure 1: Architecture.

widgets. The approach used for widget extraction is based on OpenCV[1] algorithms (e.g. *findContours*) and, the extraction of text from the original GUI is made using a Python function (*Python-tesseract*). The tool also supports text translation (see Figures 2 and 3).

An alternative approach to extract widgets from the GUI images could have been used. For example, the work of Dubrovina et al. [15] for GUI object classification using a Support Vector Machine classifier. However, we did not follow an approach using machine learning because none of the dataset found was shared.

The architecture of the approach is presented in Figure 1. It is composed by the GUIRT tool, a Virtual Machine (VM) where the original GUI is executed (with Sikuli scripts and using RunServer[2]) and hidden to the user and, the new GUI and an GUI editor (QT Designer) for fine-tuning of the automatically generated GUI.

The GUIRT tool possesses two working modes, *edition* and *execution*. The GUI redefinition is performed in the *edition* mode where the tool automatically creates from an original GUI (see example in Figure 2) a corresponding new editable GUI (see Figure 3). The creation of the new widgets is based on automatically extracted images from the original GUI that are then merged forming a new editable GUI. The result of this automatic process usually needs some fine-tuning. The QT Designer GUI editor enables developers to improve it in terms of usability, aesthetics and UX. Finally, when all GUI are fine-tuned (see one final GUI in Figure 4), GUIRT working mode can be switched to *execution*.

[1] opencv.org (last accessed May 23, 2019)

[2] SikuliX: http://www.sikulix.com/ (last accessed: May 23, 2019)

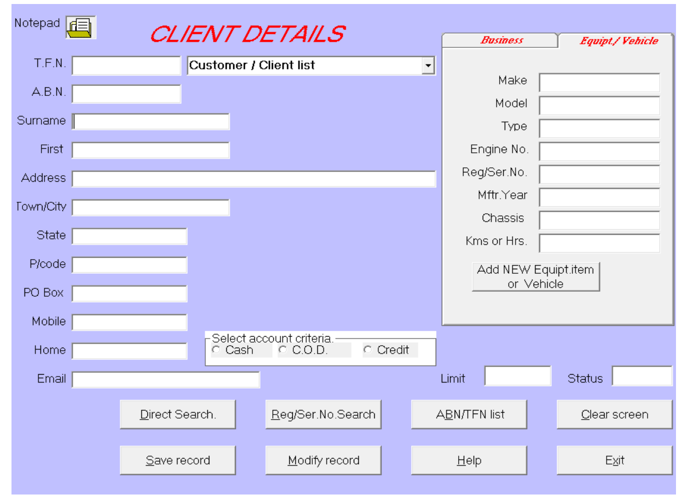

Figure 2: Original ESIBIS system - client details GUI.

Figure 3: New (automatically generated and translated to Portuguese) ESIBIS editable GUI (client details).

Execution mode (the unique provided to end-users) enables the launch of the new GUI. The logic of the application remains in the original application therefore, to reflect changes in the new GUI actions performed on it are replicated (via Sikuli script) in the original GUI (running in the VM) and corresponding results conveyed back to the new GUI. The communication of changes to the new GUI is made by automatically detecting visual changes in the original GUI. This process might take few seconds and, therefore, to mitigate potential UX effects, visual feedback (progress indicator) is provided while the process happens.

Evaluation and Results

To evaluate the results obtained with the GUIRT tool an experiment was performed applying the tool to ESIBIS[3] an old ICS. ESIBIS (see Figure 2) is a stand-alone system for full cost control and reporting of all aspects of mechanical repair, servicing or maintenance.

Participants

23 people voluntary participated in the experiment. Participants were aged between 20 and 58 years (65,2% males, 34,8% females) where 4,3% completed high-school, 65,3% obtained bachelor’s degree, 26% master’s degree and 4,3% postgraduate. Regarding Information Technologies usage, 95,7% of the participants access computer applications/websites in a daily base and 4,3% between 3 and 4 times per week.

Design

A within-subject design was used. All participant performed all tasks in both ESIBIS versions. Each participant started with the provided new ESIBIS version (with new GUIs) and then moved to the original ESIBIS version to eliminate potential carryover effects.

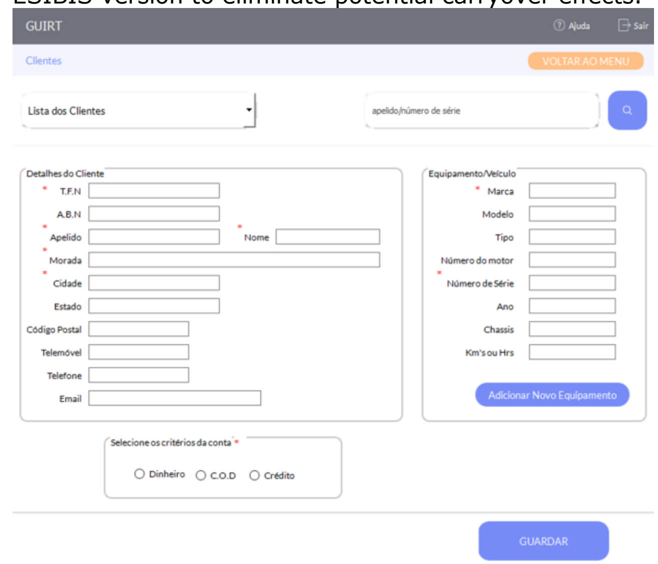

Figure 4: Example of final (fine-tuned version of the GUI presented in Figure 3) ESIBIS GUI (client details).

Procedure

The participants met the experimenter and received the procedure. Participants were not given time limit to complete the tasks. Their interaction with the ICSs was recorded. The tasks consisted in: i) creating a new client; ii) creating a new work order and, iii) finding a given client. At the end all participants filled a questionnaire that addressed four aspects (as defined in the standard USE questionnaire [16]): participant characterization, usefulness, ease of use and user satisfaction. Subjects were asked to answer on a 5-point Likert scale. The questionnaire included open question on the GUI's strong and weak points and

[3] https://archive.org/details/tucows_314329_ESIBIS

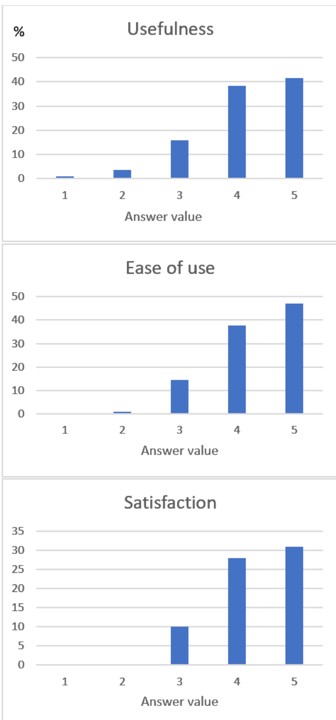

Figure 5: Questionnaire results.

enabled the participants to make any further comments they wished. The primary measures used in this study were, number of interaction mistakes made, completion time of each task and task completion success rate. The answers to the post-experiment questionnaire were secondary measures.

Results

We used a *T-test* as sample normality was verified. The new GUI led to significant reduction in the number of interaction mistakes (*p_value* = 0,0429). Differences were also identified in task completion success rate using the new GUI where all tasks where completed. With the original GUI some users where not able to complete all tasks (approx. 24% were not completed). The new GUI did not improve completion time.

The post-questionnaire (see Figure 5) indicated a positive reaction to the new GUI with all criteria obtaining a mode of 5. Overall, participants were satisfied with the new GUI. They found it useful and easy to use. The weakest aspect was the waiting time as observable in both primary measure and open answers (one participant made the following comment: "high waiting time making the user to click more than once in the button"). Participants also positively highlighted the organization, ease of use and aesthetics of the new GUI. One participant pointed out "straightforwardness of the layout and easy to use".

Discussion

The new GUI led to significant reduction of interaction mistakes. This is explained by the better design of the new GUI made possible using GUIRT. We believe that differences in the design were also responsible for improvements in task completion success.

Improvements in UX where not the focus in the example used however, the tool supports their introductions. Future evaluations with UX emphasis should be addressed.

The completion time of the tasks did not improve. This result is in line with our expectations. Data forwarding from the original GUI to the new GUI reflecting state changes takes additional time that negatively affected the performance of users while using the new GUI. Nevertheless, we believe that the reduction of mistakes made, both improved task completion success rate and acceptability (reported in the post-questionnaire) are important improvements. The fact that the redefinition if made without source code access of the original ICS makes it impossible to identify state changes in the original GUI and forward them to the new GUI without affecting performance. Nevertheless, improvements might be achievable using more efficient algorithms.

The evaluation performed was made using a stand-alone application however, GUIRT tool can also be applied to other types of applications (e.g. web application). The fact that GUIRT uses a computer vision-based approach makes it applicable to any visual interface (currently limited to a subset of widgets).

Conclusions and Future Work

This paper presented GUIRT, a tool for GUI redefinition without source code access aiming to improve usability and UX. The tool was illustrated with an example and its applicability was evaluated. Results revealed the viability of the solution, significant reduction of interaction mistakes, improvements in both task completion success rate (100% with the new GUI versus 76% with the original GUI) and user's satisfaction. Completion time did not improve.

The tool is currently limited to a subset of widgets however, improvements to consider more widgets is planned. We are currently developing features to support changes in GUIs that address extreme users' specificities in both visual and interaction levels.

Acknowledgements

Fundação para a Ciência e a Tecnologia (FCT, Portugal), projects UID/EEA/50009/2019 & UID/MULTI/0446/2013

References

1. Estelle de Kock, Judy van Biljon, and Marco Pretorius. 2009. Usability evaluation methods: mind the gaps. In Proc. of Research Conference of the South African Institute of Computer Scientists and Information Technologists. ACM, 122-131.
2. Ben Shneiderman. 2016. Designing the User Interface: Strategies for Effective HCI. Pearson.
3. FDIS, ISO: 9241-110: 2006. Ergonomics of human system interaction-Part 110: Dialogue principles. International Organization for Standardization.
4. ISO 9241-400: 2007. Ergonomics of Human-system Interaction: Principles and requirements for physical input devices. International Organisation for Standardisation.
5. Jenifer Tidwell. 2010. Designing Interfaces: Patterns for Effective Interaction Design 2nd Edition. O'Reilly Media. ISBN-13: 978-1449379704.
6. Morgan Dixon, Daniel Leventhal, and James Fogarty. 2011. Content and hierarchy in pixel-based methods for reverse engineering interface structure. In Proc. of the Conference on Human Factors in Computing Systems. ACM, 969-978.
7. Morgan Dixon, Alexander Nied, and James Fogarty. 2014. Prefab layers and prefab annotations: extensible pixel-based interpretation of graphical interfaces. In Proc. of the symposium on User interface software and technology. ACM, 221-230.
8. José Luís Silva, Jorge D. Ornelas, and João C. Silva. 2016. Make it ISI: interactive systems integration tool. In Proc. of the Symposium on Engineering Interactive Computing Systems. ACM, 245-250.
9. Gagan Singh, Sergi Badia, Rodrigo Ventura, José L. Silva. 2018. Physiologically attentive user interface for robot teleoperation - real time emotional state estimation and interface modification using physiology, facial expressions and eye movements. In Proc. of the Conf. on Biomedical Engineering Systems and Technologies, 294–302.
10. José Luís Silva, J.C. Silva. 2018. Graphical User Interface Redefinition Addressing Users' Diversity. In International Conference on Human-Centred Software Engineering, Springer, Cham, 319-326.
11. Krzysztof Z. Gajos, Daniel S. Weld, Jacob O. Wobbrock. 2010. Automatically generating personalized user interfaces with SUPPLE. Artificial Intelligence 174(12–13), 910–950.
12. Krzysztof Gajos, Jacob Wobbrock, and Daniel Weld. 2008.Improving the performance of motor impaired users with automatically-generated, ability-based interfaces. In Proc. of the Conf. on Human Factors in Computing Systems. ACM, 1257-1266.
13. Krzysztof Gajos and Daniel S. Weld. 2004. SUPPLE: automatically generating user interfaces. In Proc. of conf. on Intelligent user interfaces.ACM, 93-100.
14. Tom Yeh, Tsung-Hsiang Chang, and Robert Miller. 2009. Sikuli: using GUI screenshots for search and automation. In Proc. of the symposium on User interface software and technology. ACM, 183-192.
15. A. Dubrovina, K. Pavel, F. Daniel, S. Sagi and B. Ruth. 2012. Efficient and robust image descriptor for GUI object classification. In Proc. of Conference on Pattern Recognition. IEEE, 3594-3597.
16. A. M. Lund. 2001. Measuring Usability with the USE Questionnaire. Usability Interface, 8(2), 3-6