

Gestalt Prototyping Framework - Evaluation Tool

Daniel Ripalda¹, César Guevara¹, and Alejandra Garrido²

¹ Universidad Tecnológica Indoamérica,
Mechatronics and Interactive Systems Research Center
Machala & Sabanilla, Quito, Ecuador

² LIFIA, Facultad de Informática, Universidad Nacional de La Plata & CONICET
50 & 120, 1900 La Plata, Argentina

¹{danielripalda, cesarguevara}@uti.edu.ec ²garrido@lifia.info.unlp.edu.ar

Abstract. The Gestalt Prototyping Framework linked the Nielsen Usability Heuristics to the principles of human perception and presents some parameters that can be used in developing high and low-fidelity prototypes for mobile application interfaces. The link between the fundamentals of usability and Gestalt principles focuses on the graphical components of the interfaces and the functions that they fulfill in the development of different functional actions. Previous articles have presented promising experimental results in reducing trial-error regressions in the interface design process and improving interface redesign processes; There have been positive results in parameters such as learnability, ease of use, perception of simplicity, and user preference. Within this same line of research, the present work describes the development process of an application that summarizes the fundamental aspects of the Gestalt prototyping framework in an evaluative model, which helps application designers, software engineers, and usability experts, to assess the prototypes of the interfaces and the incidence of the different graphic components in usability interactions. The tool is based on the most widely used usability parameters and is structured based on the responses that the development team fills in an automated LIKERT assessment. The results are processed based on an algorithm that simulates and predicts the results that could be obtained in high-fidelity prototype tests. This application issues specific recommendations on the visual components of the interfaces, to obtain better results in the production of high and low-fidelity prototypes.

Keywords: Gestalt, User Interfaces, Usability, Mobile, Agile Software Development.

1 Introduction

The Gestalt Prototyping Framework links the Nielsen Usability Heuristics with the principles of human perception and presents a series of parameters that can be used by designers, software developers, and usability experts, during the development stage of high and low fidelity prototypes. in the production of mobile applications. A link was found between the statements of the heuristics, the graphical components of the interfaces, the principles of human perception of Gestalt, and some interactions of the user experience.

The first stages of this research describe the Framework from a constructive model that guides the Development Team in the production of mockups, providing specific guidelines for the use of iconography, menu configuration, button interaction and content organization, factors that affect the learnability, memorability, efficiency, and usefulness of an interface; the relationship between visual components and interactions facilitate that a heuristic can be understood from specific visual representations, reducing the developer's margin of interpretation in the early stages of a project, without interfering with the methodology used in the project. In the experimental phase of the constructive model, he obtained promising results in the redesign of an interface prototype, increasing the perception of speed and use by 33.1% and 97% in preference of use, having measured experience, learnability, and ease.

This paper describes in section 2 some works related to usability measurement tools, section 3 describes the concepts that were used to structure the application, section 4 describes the development process and general operation, and section 5 describes the results of tests performed during application production.

2 Related work

With the rapid development of the mobile application market, there has been a notable interest in developing tools to measure different criteria in the Interfaces of mobile devices, one of the most interesting systems is described in Automated model-based Android GUI Testing, using multi-level GUI comparison criteria [1], this tool allows checking the bugs of an application by configuring test cases based on the actions that occur in an interface and could be used to test high-quality models.

Another tool is integrated automated test case generation for safety-critical software [2], which is a patented tool that among its various functions can generate GUI tests and verify the usability of an industrial application. This program is based on a device that obtains the requirements of the users, generates an intermediate model, and prepares test cases for an application, This work has made it possible to obtain important data on the hardware functions of the devices and the instructions of the program that controls it. Other programs [3] use this methodology to detect errors in the requirements, using an analysis module that compares other requirements to detect conflicts and repetitions.

One of the most recent revisions regarding this type of application was developed by Azham Hussain [4], In his work, ten different applications are reviewed such as Perfect Mobile, Device Anyware, Monkey Runner, Robotium, among others. According to this author, one of the tools with the best performance is Micro Focus Silk Mobile, a Micro Focus product; This multi-platform application performs tests based on the recognition of images of the interface and captures information about the performance of the device, validating the quality of the construction of the software, it can save, process and simulate user interactions with generating detailed reports that are later analyzed.

Another quite interesting tool is HUI Analyzer, which was reviewed by Simon Baker [5], The tool is capable of interpreting data related to Learnability, Understandability, Efficiency, Memorability, and Satisfaction; and it can find different defects in the

usability of an application using mathematical models; The technical documentation available explains that the quantitative evaluation scheme could minimize the effectiveness of the results since, for example, the contrast factor of the interface is difficult to evaluate because at the same time that it can increase performance, it can compromise aesthetics.

In general, the tools to measure heuristics still have limitations, in the article Finding Usability Problems Through Heuristic Evaluation [6], Nielsen points out that some of the guidelines are abstract and leave room for interpretation by usability experts using their criteria and personal experience. Some studies show that automated tools can be 50% effective in detecting usability problems due to false positives [7]. However, in some experimental tests [8] in applications such as pCloudy, Test Object, Device Anywhere, and Perfect Mobile, they obtained an 85% effectiveness in identifying different types of failures, such as application crashes, failures in the loading of interfaces, or incomplete information in the display; although these applications are quite effective in detecting functional problems, we cannot say that they detect errors or make recommendations in aspects directly related to usability.

3 Concepts applicable to the evaluation tool

The Gestalt Prototyping Framework [9], It has an evaluative model that is based on the idea that the graphic components of the interfaces affect specific usability interactions so that the heuristics can be perceived; therefore, if the perceptual fundamentals are used in these components, they can be associated with the heuristics, providing information during the low and high fidelity prototype development phase.

The usability heuristics [10] are seen as guides to develop the functional interactions of the applications, while the Gestalt [11] provides guidelines to develop the visual aspect. Using conditional logic to structured tools [12], these concepts can be used to evaluate prototypes and detect errors in components, predict errors in usability tests, and measurement of aesthetics. Figure 1 explains the concept of the evaluative model of the Framework applied to a software tool.

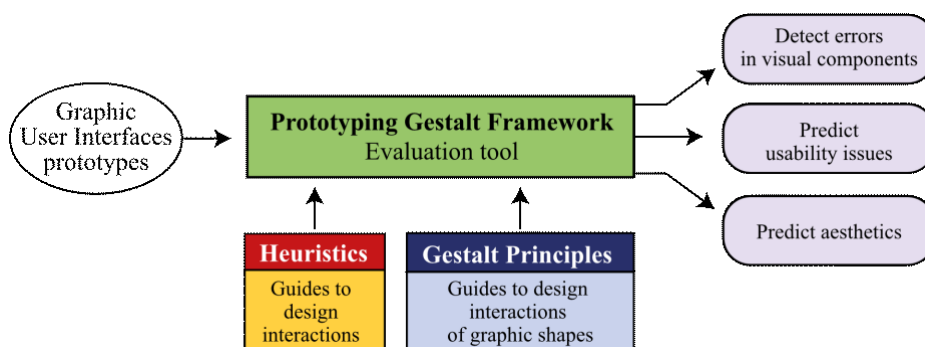


Fig. 1. Concepts of the Gestalt Prototyping Framework applied to an evaluative tool

According to Baker's analysis, this tool is a model-based Guideline Checker [13], focused on evaluating the interfaces for specific usability parameters; during the product development phase of a mobile application, the usability experts can use it to interpret qualitative data in a static analysis [14], that detects problems relating the visual aspect of the GUI with the functional properties of the application.

4 Application development

A team of students and teachers from the Indoamérica University developed the application taking into account the established parameters, it is a mobile application for the Android platform, two types of tests can be executed, one focused on specific components of the interfaces, and the other that covers a complete test. The application asks a series of questions related to the graphic components of the mobile application interfaces to be evaluated, the usability expert answers the questions through a Likert scale, the answer to each question assigns a score to the heuristic and the specific usability interaction; for example, in the iconography category, the question “Icons can be easily recognized?” assigns a point to the heuristic “Match between system and the real world”, at the same time assigns a point to the interaction “Learnability” and another point to “Efficiency”. The perception of aesthetics is calculated based on the statement that there is a correlation between usability, the perception of aesthetics [15], and credibility [16]; therefore, the aesthetic assessment is given based on the general score assigned to each component. The results are quantified according to Table 1

Table 1. Relation of Interface components, heuristics, and usability interactions

Questions related to	The score is assigned to the heuristic	The score is assigned to the usability Interactions
Iconography Colour Buttons	Match between system and the real world	Learnability Efficiency
Menus Submenus Pop up Slide buttons	Recognition rather than recall	Learnability Efficiency Memorability
Backgrounds Dividers Graphic style	Aesthetic and minimalistic design	Memorability Easy

Once the user completes the test, the system performs a calculation to quantify the results by heuristics, by usability interaction, and by the perception of aesthetics; To present the final values, an algorithm is used to adjust the results to the parameters of real usability tests carried out by human users.

5 Results

The name that was selected for the Project and the application is GEPROF-EMAT, (Gestalt Prototyping Framework - Evaluative Mode Automated Tool), the beta version has been developed with the SCRUM methodology, using Unity technology, for the Android platform and run locally. In the preliminary tests, it was possible to see that the calculation carried out by the application is consistent with the applied concepts and the guidelines of the Framework. In the evaluation of prototypes, it was possible to appreciate positive evaluations around some parameters that have been measured in similar applications, according to the parameters evaluated by Baker, the application seem to meet expectations such as: Not being intrusive to the user, providing sufficient information to perform static analysis and maintaining an independent evaluation process. Figure 2 shows the main application interfaces



Fig. 2. Application interface

6 Conclusions and future work

The application GEPROF-EMAT is in the testing phase, has been complying with the different interactions typical of the development methodology that has been applied, the first results are encouraging and there is a projection that the margin of error for tests with real users is currently 35%, this value will decrease as continue using the tool and comparing the results to make adjustments to the prediction algorithm. It is important to point out that this tool even in its final version will not replace usability tests or the current measurement instruments, because just provides referential information for the construction of prototypes and establishes approximate values of the real tests. The quantitative assessment of the tool prompts the discussion of the development team of a mobile application, the results require human intervention and interpretation, as a determining factor in identifying and correcting errors. The developer's vision, intuition, background, and experience are required to positively validate the interfaces and find functional solutions that consider abstract aspects such as aesthetics. We expect that in the near future, the effectiveness of the application is expected to reach 85%, equaling automatic error detection tools; for this, it is necessary to make adjustments in the error prediction algorithm based on the values obtained in usability metrics measured with real users

7 References

- [1] Baek, Y. M., & Bae, D. H., Automated model-based Android GUI testing using multi-level GUI comparison criteria, Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, 2016, pp. 238-249.
- [2] LI, Meng., Integrated automated test case generation for safety-critical software, U.S. Patent No 10,108,536, 2018.
- [3] Manolios, P., U.S. Patent No. 9,639,450, Washington DC: Patent and Trademark Office, 2017.
- [4] N. H. Saad, The perceived usability of automated testing tools for mobile applications., 12 ed., Journal of Engineering, Science and Technology (JESTEC), 2017, pp. 89-97.
- [5] Baker, S., Au, F., Dobbie, G., & Warren, I., Automated Usability Testing Using HUI Analyzer, 19th Australian Conference on Software Engineering (aswec 2008), 2008.
- [6] Nielsen, J, Finding Usability Problems Through Heuristic, Monterrey, California USA: In proceedings of the SIGCHI conference on, 2002.
- [7] Dix, A, Finlay, J., Abowd, G. D., and Beale, R., Human-Computer Interaction, Harlow, England: Prentice-Hall, 2004.
- [8] Vilkomir, S., Marszalkowski, K., Perry, C., & Mahendrakar, S., «Effectiveness of multi-device testing mobile applications.» *ACM International Conference on Mobile Software Engineering and Systems*, pp. 44-47, 2015.
- [9] Ripalda, Daniel; Garrido, Alejandra; Guevara, César, «Gestalt framework in the design of interfaces for mobile devices: Theoretical approach.» de *13th Iberian Conference on Information Systems and Technologies (CISTI)*, Cáceres, Spain, 2018.
- [10] Nielsen, J., & Molich, R., «Heuristic evaluation of user interfaces.» de *In Proceedings of the SIGCHI conference on Human factors in computing systems. ACM*, 1990.
- [11] Hartmann, G. W., «Gestalt psychology: A survey of facts and principles.» 1935.
- [12] Vessey, I., & Weber, R., «Structured tools and conditional logic: An empirical investigation.» *Communications of the ACM*.
- [13] Beine, M., «A model-based reference workflow for the development of safety-critical software.» 2010.
- [14] Engler, D., & Musuvathi, M., «Static analysis versus software model checking for bug finding.» *In International Workshop on Verification, Model Checking, and Abstract Interpretation*, pp. 191-210, 2004.
- [15] Frison, A. K., Wintersberger, P., Riener, A., Schartmüller, C., Boyle, L. N., Miller, E., & Weigl, K., UX We Trust: Investigation of Aesthetics and Usability of Driver-Vehicle Interfaces and Their Impact on the Perception of Automated Driving, Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019.
- [16] Oyibo, K., Adaji, I., & Vassileva, J., Mobile Web Design: The Effect of Education on the Influence of Classical and Expressive Aesthetics on Perceived Credibility, International Conference on Human-Computer Interaction, 2019.