Comparative Evaluation? Yes, But With Which UI?

User's feedback offers valuable information that help designers to improve their work. In this paper, we present a study on user's feedback when evaluating a User Interface (UI) by comparison. Our aim is to define the properties that the alternative UI must satisfy to maximize the benefits of comparative evaluation.

The user interfaces proposed for the evaluation were designed using the CAMELEON Reference Framework (CRF) and covering variations at each level of abstraction. We studied the impact of each variation on the users' feedback. We found that when the alternative design refers to the same Task Model (TM) as the original one but has a different AUI (Abstract User Interface), the number of negative returns is significantly higher, making the comparative evaluation more productive.

HCI. UI Design. UI Evaluation. Feedback. CAMELEON reference framework.

1. INTRODUCTION

Feedback is very important for the design process. Critiques, opinions and suggestions are valuable information to improve the design (Nguyen, 2017), (Hui, 2015). One of the most famous saying by B.Gates is "*We improve our products, based on feedback, until they're the best*". Traditionally when asking for users' feedback, designers present only one interface, *the* one to be tested. However, as demonstrated in (Tohidi, 2006) providing several design alternatives to the assessor increases the amount of feedback and facilitates comparative reasoning. However, to the best of our knowledge, there is not yet any research studying the characteristics that the alternative design must satisfy in order to maximize the benefits of the evaluation.

This research aims to improve the comparative assessment by producing the optimal alternative design, i.e. maximizing returns to the original User Interface. In this study, we conducted an experiment, where we use the CAMELEON Reference Framework to generate User Interfaces depending on different classes of variations of a design and study their impact on users' feedback through a comparative evaluation. This study will help us to define the criteria that the alternative design must meet to maximize feedback on the interface of interest.

2. RELATED WORK

2.1. Testing many is better than testing one

Working with examples has proven to have several benefits for both the learning process and the outcome (Lee, 2010). Accordingly, designers often use examples for inspiration, which offers contextualized illustrations of how form and content integrate. According to (Herring, 2009), examples are crucial to design activities. They support both the generation of new ideas and the selection of interesting ones. Examples enable to identify limitation of previous designs and reinterpretation and recombination of ideas (Masson, 2011).

Besides using alternative designs and examples during the design process, it has been proved that using multiple designs can also improve the results of the design evaluation. Creating multiple prototypes facilitates comparative reasoning, grounds team discussion, and enables situated exploration (Tohidi, 2006).
In (Wiklund, 1992), Wiklund et al. studied the impact of the fidelity of software prototypes on the perception

of usability. The result of their research lead to this observation:

> In studies such as this one, we have found subjects reluctant to be critical of designs when they are asked to assign a rating to the design. In our usability tests, we see the same phenomenon even when we encourage subjects to be critical. We speculate that the test subjects feel that giving a low rating to a product gives the impression that they are "negative" people, that the ratings reflect negatively on their ability to use computer-based technology, that some of the blame for a product's poor performance falls on them, or that they don't want to hurt the feelings of the person conducting the test.

Dicks et al. Research in (Dicks, 2002) show that when people are shown multiple prototypes, they could feel less pressured to impress the experimenters by praising a particular design. Being presented with multiple alternative designs may allow for a more accurate comparative evaluation.

In (Tohidi, 2006), Tohidi et al. examined the differences that would occur between a usability test that exposed users to a single design, and one where they were exposed to three different alternatives. This study showed that designs are rated higher when seen alone than they would be when seen in comparison with other designs. Additionally, the number of designs given to evaluate, can influence the quantity, quality, and responsiveness of the feedback.

However, in their study, there were no justifications or explanation of the choice of the alternative designs given to users during the usability test.

In our work, we tried to test the impact of the alternative design given during the comparative evaluation on the user's feedback. We believe that the choice of this UI could remarkably influence the feedback received from users.

2.2. CAMELEON reference framework

CAMELEON is a Reference Framework for User Interfaces in Multiple Contexts. It decomposes user interface design into a number of different components that seek to reduce the effort in targeting multiple contexts of use (Calvary, 2003).
This framework structures the development process within four levels of abstraction (*Figure1*):

- **Task and domain** is the top level that describes the various interactive tasks which can be realized by the end user and the domain objects that are manipulated by these tasks,
- **Abstract User Interface (AUI)** makes design decision about grouping and navigation,
- **Concrete User Interface (CUI)** makes design decisions about rendering. It defines how the UI is perceived and can be manipulated by end users,
- **Final User Interface (FUI)** is the running UI. Design decisions are about the programming or mark-up language to be used to represent the UI.

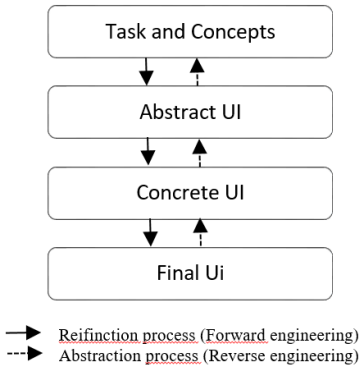

Figure 1. *A simplified version of the CAMELEON Reference Framework*

These four levels are structured with a relationship of reification (going from an abstract level to a concrete one) and/or abstraction (going from a concrete level to an abstract one) (Calvary, 2003).

3. PRELIMINARY STUDIES

We conducted an experiment in which participants had to evaluate different designs. Each design is generated depending on a different variation of the first three abstract levels of CRF: the task and domain level, the AUI level and the CUI level.

Our goal in this study is to identify which generated interface influences the user's feedback the most and then to define the criteria that this interface must meet to maximize returns on the interface produced by the designer.

We assume that these variations affect the user's feedback as follows:

H1: The choice of the alternative design conditions the user's return to the original one.

H2: The comparative evaluation will be more productive with a design that has the same task model but a different abstract UI.

3.1. Method

We started by creating different designs for the same application following the CRF architecture so we would have various alternatives of the same UI. In order to do that, we first defined the variations of each abstract level of CRF except the final UI level as followed:

- **Task and concept:**

  Model structuring: e.g. factorization
  Operators between tasks: e.g. replace the choice operator by a sequence one
  Task Decorations: e.g. decree a frequent task

- **Abstract UI:**

  Groupings: e.g. group all frequent tasks to separate them from non-frequent ones.
  Navigation: e.g. launch the application on the space of frequent tasks and force non-frequent tasks to go through frequent tasks.

- **Concrete UI:**

  Integrators: modality (graphic), widgets (radio buttons, check box…)
  Settings: colour, size, position...

We only defined the characteristics for generating variants for the first three abstraction levels because we only need a simple paper prototype for the evaluation. The Final UI implies coding which will be a waste of time. This way, the prototypes are quick and inexpensive to make.

3.2. Case study

We chose to design an application that allows the user to check and manage the security of their house remotely. The motivation is that such an application is practical and widespread.
Three main tasks were proposed to the user: **(1)** control the access to the house, **(2)** control the security cameras and **(3)** manage the alarm system. Controlling the access to the house allows the user to remotely lock or unlock the doors and other entries of the house. The user can watch the feed from the security cameras (in real time or recorded), send or delete them. Finally managing the alarm system allows the user to program it and to stop the alarm when triggered.

3.3. Design proposals

The Task models below (figure 2 and figure 5) were designed using « Flexilab », a multimodal editor illustrated on task modeling created by (N. Hili, 2015). Figures 3 and figure 4 present two alternatives of AUI related to Task Model 1. Figure 6 proposes an AUI for the Task Model 2.

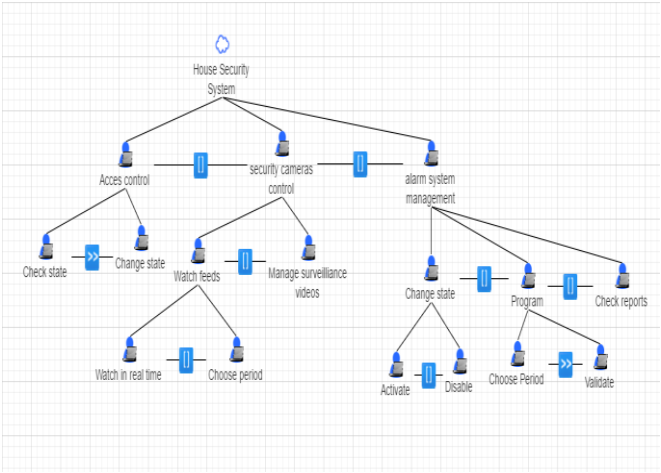

***Figure 2**. Task Model 1*

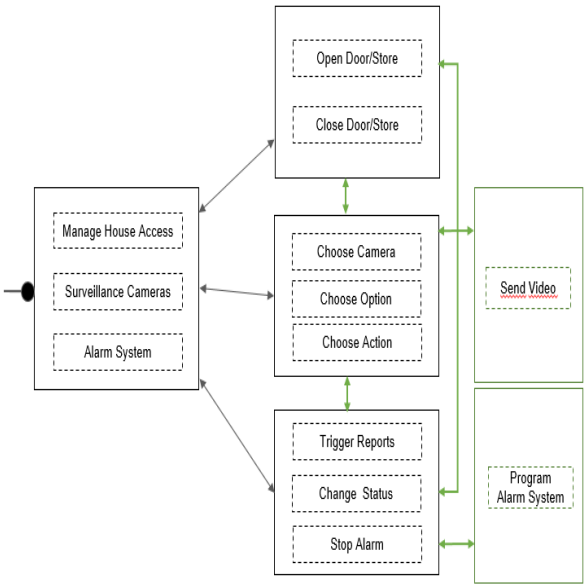

***Figure 3.** AUI_1*

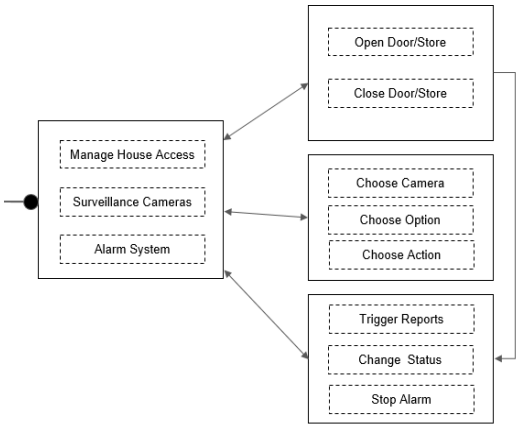

Figure 4. *AUI_2*

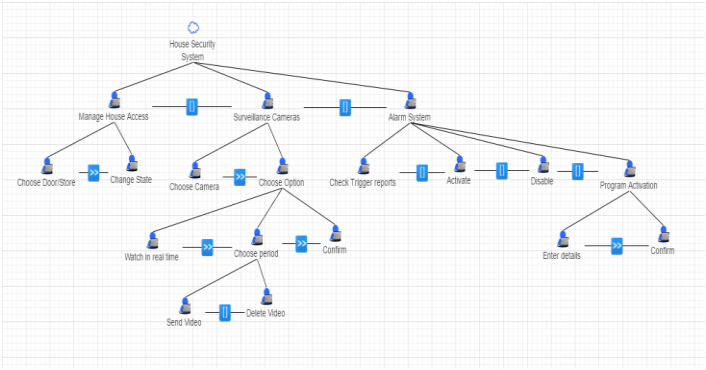

Figure 5. *Task model 2*

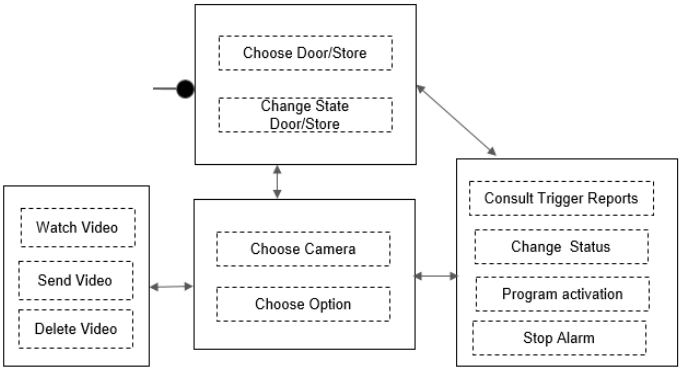

Figure 6. *AUI_3*

3.4. Participants

We had a total of 28 participants: PhD students, recent graduated doctorate, master students, engineers and other students in computer science.

4. EXPERIMENT

We first started by briefly introducing the study. We then explained the security house application and its main functionalities. Finally, we gave each participant two designs and a questionnaire. We did not enforce any time limit; the participants took their time to observe each UI.

The questionnaire given to each participant within the two designs is divided in two parts. The first part was composed of 11 questions based on a 5-point Likert scale. This first part is meant to rate the design. These questions were about three main aspects: Content (organization), navigation (structure/navigation tool) and design (visual). In the second part of the evaluation, the participants were asked to give their opinions concerning these six dimensions of design evaluation: navigation, aesthetics, readability, consistency, exportability and learnability.

We divided the participants into 3 groups. The first group was asked to evaluate UI_1 with UI_2. The second group evaluated UI_1 with UI_3, and the third group evaluated UI_1 with UI_4. The Interfaces were selected as presented below:

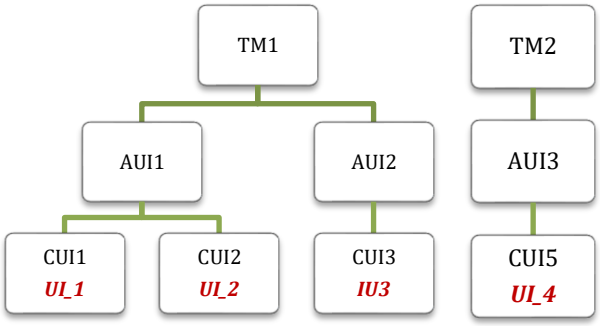

Figure 7.*Choice of the alternative design*

The aim of this experiment is **(1)** to see which variant gives more feedback, and to **(2)** see the impact of each abstraction level variations on the user's feedback. In order to define which alternative design to use during interface evaluation.

5. EVALUATION RESULTS

5.1. Categorization of user's feedback

In order to classify the users' statements (critique, opinion and suggestion), we used the taxonomy elaborated in (Tohidi, 2006) but we adapted it to our needs. In their work, Tohidi et al. divided the user's

statements as shown in *Figure 8* where comments are facts or personal opinions about the design. **Suggestions** are propositions for change to improve the current design. The comments were either **positive** or **negative**. As for the suggestions there were classified as **substantial** or **superficial**. In terms of the substantial suggestions: ideas for improvement that were original (**new**) or **borrowed** from ideas they had seen in other interface.

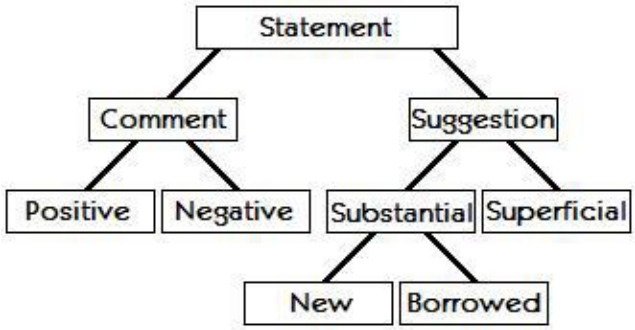

Figure 8. *Categorization of User Feedback according to Tohidi et al.*

In our case, we only had two major types of statement "comments" or "suggestions". The comments are classified as either "positive" (Easy and convenient navigation), or as "negative", (there is too much information in the interface).

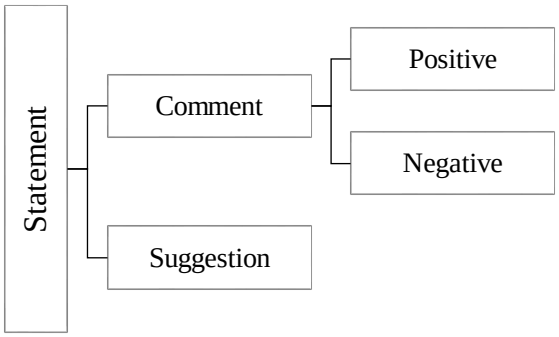

Figure 9. *Categorization of User Feedback*

5.2. Impact on user ratings depending on the choice of the alternative design

In order to assess the impact of comparing UI_1 to different designs (same task model but different AUI; same task model and AUI but different CUI, different task model), we first calculated the average overall score of UI_1 based on the participant's rating given in the questionnaire.
Then, for each different prototype, we compared the score assigned to UI_1 when it was seen with each of the other design. Finally, we calculated the number of statements about UI_1 each time when given with a different UI.

The first observation was that the choice of the second UI conditions the user returns and opinion to the first one. For example, when comparing UI_1 to UI_2, a user did not comment about the interface navigation, but when comparing UI_1 to UI 3, the user started criticizing the navigation or the widgets.

The number of statements when comparing UI_1 to UI_3 was higher than when comparing UI_1 to UI_2, and UI_4. Also, the average score given to UI_1 when seen with a design with a different AUI was lower than when seen with the other designs (see *Table1*).

These results support our hypotheses**: (1)** the comparative evaluation is more productive with a UI that has the same task model but a different abstract UI. **(2)** The number of suggestions to improve is significantly higher when comparing UI_1 to UI_3 than when comparing UI_1 to UI_2 or UI_4.

An observation that we did not expect, is that when evaluating the UI with one design that has a different TM, the number of positive feedback is significantly lower.

Evaluation / Statement type	UI_1 seen with UI_2	UI_1 seen with UI_3	UI_1 seen with UI_4
Number of statement	18	**29**	17
Positive comments	5	**4**	2
Negative comments	4	**14**	8
Suggestions	9	**11**	7
Average score	0.68	**0.58**	0.63

Table 1. *Impact on user ratings depending on the choice of the alternative design*

6. CONCLUSION AND FUTURE WORK

As demonstrated in (Tohidi, 2006), user's feedback is affected by the number of design alternatives they are exposed to. In this paper, we conducted a study to observe the user's feedback depending on the

alternative UI given for comparison. The aim of this research is to define the criteria that the alternative UI must meet to maximise returns on the original one. We used the CRF to characterise the UIs variations.
We observed that the user's opinion about the UI under study was remarkably affected by the choice of the alternative design presented. Analysing the feedback, we found that the AUI variants affect the users' feedback the most in terms of rating (score) and the number of statements.
In the next step we will explore further criteria, and once well defined, we will develop a tool for generating the *best* UI for supporting comparative evaluation at low-cost and for high-benefit.

7. REFERENCES

A. Xu, Brian P. Bailey.2012. What Do You Think? A Case Study of Benefit, Expectation,and Interaction in a Large Online Critique Community. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*.295-304

B. Lee, Savil Srivastava, Ranjitha Kumar, Ronen Brafman, Scott R Klemmer.2010. Designing with Interactive Example Galleries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (CHI '10). 2257-2266

ConcurTaskTrees. Retrieved April 2017 from https://en.wikipedia.org/wiki/ConcurTaskTrees.

Dicks, R. S.2002. Mis-Usability: On the Uses and Misuses of Usability Testing. *In Proceedings of* the 20th annual international conference on Computer documentation.

D.Masson, Alexandre Demeure, Gaelle Calvary.2011. Examples Galleries Generated by Interactive Genetic Algorithms .In *procedings of the Second Conference on Creativity and Innovation in Design.* 61-71

D. T. Nguyen, Thomas Gancarz, Felicia Ng, Laura A. Dabbish, Steven P. Dow.2017. Fruitful Feedback:Positive Affective Language and Source Anonymity Improve Critique Reception and Work. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (CSCW '17).1024-1034

G.Calvary Joëlle Coutaz, David Thevenin, Q. Limbourg, L. Bouillon, Jean Vanderdonckt.2003. A Unifying Reference Framework for multi-target user interfaces.Interacting With Computers Vol. 15/3 . 289-308.

G.John, S. Design Prototypes.1990. A Knowledge Representation Schema for Design.AI Magazine: Volume 11 Issue 4, Winter 1990.

G.Meixner, Gaëlle Calvary, Joëlle Coutaz Introduction to Model-Based User Interfaces Retrieved January 2014 from https://www.w3.org/TR/mbui-intro/.

S.R. Herring, C.C. Chang, J. Krantzler, and B.P.Bailey .2009. Getting inspired!: understanding how and why examples are used in creative design practice. In *Process of the 27th international conference on Human factors in computing systems*. (ACM 2009). 87-96.

J. Manuel, Juan M. González Calleros,Gerrit Meixner, Fabio Paternò, Jaroslav Pullmann, Dave Raggett, Jean Vanderdonckt. Model-Based UI XG Final Report. Retrieved on April 2010 from https://www.w3.org/2007/uwa/editors-drafts/mbui/Model-Based-UI-XG-FinalReport.html.

J. Hui, Amos Glenn, Rachel Jue, Elizabeth Gerber, Steven Dow. 2015. Using Anonymity and Communal Efforts to Improve Quality of Crowdsourced Feedback.In P*roceeding of Third AAAI Conference on Human Computation and Crowdsourcing* (HCOMP-15)

K. Luther, Amy Pavel, Wei Wu, Jari-lee Tolentino, Maneesh Agrawala, Björn Hartmann, Steven Dow 2014. CrowdCrit: Crowdsourcing and Aggregating Visual Design Critique. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing.* 21-24

M. Tohidi, William Buxton, Ronald Baecker, Abigail Sellen. 2006. Getting the Right Design and the Design Right:Testing Many Is Better Than One. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (*CHI'2006*).1243-1252*

M. Tohidi, William Buxton, Ronald Baecker, Abigail Sellen. 2006. User Sketches: A Quick, Inexpensive, and Effective way to Elicit More Reflective User Feedback. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles (NordiCHI'2006).105-114*

N. Hili, Y. Laurillau, S. Dupuy-Chessa, G. Calvary. 2015. Innovative Key Features for Mastering Model Complexity: FlexiLab, a Multimodel Editor Illustrated on Task Modeling. *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems.*

Wiklund, M., Thurrott, C., and Dumas, J. 1992. Does the Fidelity of Software Prototypes Affect the Perception of Usability? In *procedings of*. Human Factors Society 36th Annual Meeting, (1992), 399-403.