HAL open science

Embedding Requirements in Design Rationale to Deal Explicitely with User eXperience and Usability in an "intensive" Model-Based Development Approach

Célia Martinie, Jean-François Ladry, David Navarre, Philippe Palanque, Marco Winckler

To cite this version:

Célia Martinie, Jean-François Ladry, David Navarre, Philippe Palanque, Marco Winckler. Embedding Requirements in Design Rationale to Deal Explicitely with User eXperience and Usability in an "intensive" Model-Based Development Approach. Workshop on Model Driven Development of Advanced User Interfaces 2010, Apr 2010, Atlanta, Georgia, United States. pp.29-32. hal-03651212

HAL Id: hal-03651212

https://hal.science/hal-03651212v1

Submitted on 26 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Embedding Requirements in Design Rationale to Deal Explicitly with User eXperience and Usability in an "intensive" Model-Based Development Approach

Célia Martinie, Jeff Ladry, David Navarre, Philippe Palanque & Marco Winckler
IRIT - University Paul Sabatier
118, route de Narbonne, 31062 Toulouse Cedex 9, France
{martinie, ladry, navarre, palanque, winckler}@irit.fr

ABSTRACT

Requirements engineering for interactive systems remains a cumbersome task still under-supported by development processes. Indeed, in the field of HCI, currently the most common practice is to perform user testing to assess the compatibility between the designed system and its intended user. Other approaches such as scenario-based design [14,16], promote a design process based on the analysis of the actual use of a technology in order to design new technologies better supporting users' tasks. However, these approaches do not provide any support for a) the definition of a set of requirements that have to be fulfilled by the system under design and b) as a consequence for assessing which of these requirements are embedded in the system and which ones have been discarded. This position paper proposes a notation and a tool for addressing precisely these two challenges. These elements are integrated within a more global approach aiming at providing notations and tools for supporting a rationalized design of interactive systems following a model-based approach.

Keywords

Model-based UI design, requirements, interaction technique, design rationale, user experience, usability.

INTRODUCTION

Traceability of choices and systematic exploration of options is a critical aspect of the development processes in the field of safety critical systems. Some software standards such as DO 178 B [15] (which is widely used in the aeronautical domain) require the use of methods and techniques for systematically exploring design options and for increasing traceability of design decisions. However, such standards only define what *should* be done and provide no information on *how* such goals can be reached by designers. Recent work in the field of software engineering has been trying to provide solutions to that problem and a collection of papers on that topic can be found in [4]. One of the remaining problems pointed out by many contributions, such as chapters 1, 19 and 20, is that requirements are poorly or even not addressed. However, Requirements Engineering (which is the first phase of the development process) provides input to all the subsequent phases and must be dealt with adequately. Indeed, ESARR (Eurocontrol Safety Regulatory Requirement) on Software in Air Traffic Management Systems [5] explicitly requires traceability to be addressed in respect of all software requirements (p. 11 edition 0.2).

This position paper addresses the problem of traceability of requirements for model-based approaches. It tackles the problem by providing an extension to a notation TEAM and its associated tool DREAM which have previously been presented in [9]. The current contribution makes it possible to relate design options with functional and non functional requirements. While other approaches such as SCRAM [16] focus on requirements identification, our approach is intended for supporting the traceability of such identified requirements within interactive systems models. As an example we show how two different interaction techniques modeled using ICOs [11] satisfy different functional and non-functional requirements. While the approach could address any kind of requirements, we put the emphasis on *Usability* and *User eXperience*.

Next section quickly presents the basic principles of the TEAM notation and the extensions that have been made to include information related to requirements. They will then be used in the third section on a case study for comparing the two interaction techniques with respect to requirements providing ways of answering two fundamental questions: 1) Which current design (among the many ones available following a prototyping phase for instance) satisfies a given requirement and 2) What is the exhaustive list of requirements fulfilled by a particular design.

EXTENSION OF THE TEAM MODEL AND NOTATION

The TEAM notation (Traceability, Exploration and Analysis Method) and its CASE tool DREAM (Design Rationale Environment for Argumentation and Modeling) has been proposed in [9] to support the exploration of options and traceability of choices during the development process of interactive safety critical systems.

TEAM notation

The TEAM notation is based on Question Option Criteria Design Rationale notation introduced by MacLean and al. [10]. QOC notation allows to list the available options for a design interrogation and to trace the selection of an option with regards to a list of relevant criteria. The TEAM notation is an extension of QOC that enables the recording

Pre-proceedings of the 5th International Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI 2010): Bridging between User Experience and UI Engineering, organized at the 28th ACM Conference on Human Factors in Computing Systems (CHI 2010), Atlanta, Georgia, USA, April 10, 2010.

Copyright © 2010 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. Re-publication of material from this volume requires permission by the copyright owners. This volume is published by its editors.

(in an exhaustive manner) of much information produced during design meetings. Such information can be:

- The questions that have been raised,
- The design options that have been investigated and the ones that have been selected,
- The evaluation performed for the different options,
- The collection of factors that have been taken into account and how they relate to evaluation criteria,
- The task models corresponding to options
- The resulting scenarios

Besides this recording of information, an important feature is to record design decisions and relate them to selected factors.

This notation and associated tool can then leverage the design rationale process for several interactive applications and help engineers in deciding to reuse or not conception choices when facing an already experienced issue.

Adding requirements to the TEAM notation

In the earlier version, the notation did not allow to express the needs and requirement. But this is required from the designers' extensive work to trace back for the selected options what were the requirements met. In addition, this lack of relationship prevents designers from exploiting requirements for the generation of options and/or to take into account requirements aspects when designing an option. We have thus added requirements as an explicit entity in the TEAM notation and in the DREAM tool. Several requirements are represented in Figure 1 (grey rectangles with a folded corner on the top left hand side) but the content of the figure will only be described in the next section. As far as HCI aspects are concerned this addition is very important as it makes it possible to explicitly represent contributing factors to usability and User eXperience (UX) and thus assess the relevance of design options to them. The requested usability requirements relate to the efficiency and effectiveness factors of the system from an ISO 9241-11 perspective [8]. The requested user experience requirements relate to the pragmatic quality and stimulation hedonic quality factors of the system from a Hassenzahl perspective [6]. This specific aspect will be detailed in the next section through a case study.

CASE STUDY

The application we chose allows a user to remove a set of icons on a computer screen. It has first been used and presented by Maurice H. ter Beek and al. to evaluate the "*Resilience of interaction techniques to Interrupts*" [17]. In order to represent design choice and design rationale we consider two different interaction techniques for performing that task. The first interaction technique only uses a mouse-based one while the second is multimodal as it uses speech additionally. The first interaction technique is an enriched drag and drop interaction, with which the user is able to move an icon onto the trash icon. When the system detects that the icon file has entered a 2 centimeters circle area around the trash icon, the user has 2 seconds to drop the file icon on the trash, or the trash icon will move to another location on the display. More sophisticated techniques could be easily modeled using our approach but this is beyond the scope of the position paper. The second interaction technique is a speech and click interaction where the user utters "delete" and clicks on the icon to be deleted.

The *Speech & Click* and *enriched Drag & Drop* options, among other design choices, are going to be evaluated according to the initial requirements for the application. One having an important impact is the one requiring the interaction technique to be tolerant to interruptions (for more details see [17]).

Presentation

An initial list of these requirements gathers functional needs and non-functional ones (mainly Usability and User eXperience). An extract of the set of usability requirements for this user interface are ("u" stands for usability requirement):

- Ru1-The application shall support one interruption every 10 seconds
- Ru3-The application shall allow the user to clean the desktop in less than 30 seconds

The set of user experience requirements for this user interface are ("ux" stands for user experience):

- Rux1-80% of the users shall find the application easy to use
- Rux2-80% of the users shall find that interacting with this application is surprisingly different

Given this list of requirements, evaluation criteria are defined and linked to appropriate factors. The different questions, options and design paths with regards to elected criteria are consigned within the DREAM design rationale tool. Figure 1 gives an overview of the design options linked to requirements and evaluation criteria. Due to space constraints, the schema has been shrunk, but several display techniques for the tool are currently being studied for big and complex diagrams. Rectangles show the requirements, rounded-shape squares contain the design questions that have been asked, right-angle triangles on the right side describe factors and the other triangles describe criteria. Each question has several possible options to address the issue. For instance, as far as the interruption is concerned ("How to display interruptions?"), the interruption can be displayed as a "pop-up window" or as a "small icon blinking" on one side of the display. In the TEAM mode, the connection between these options and the four criteria "File deletion error rate", "Time to clean desktop", "Perceived manipulability" and "Perceived originality" represents the relative impact of the option. The strong link between the option "small icon blinking" and the "time to clean desktop" criterion shows that it favors that criterion.

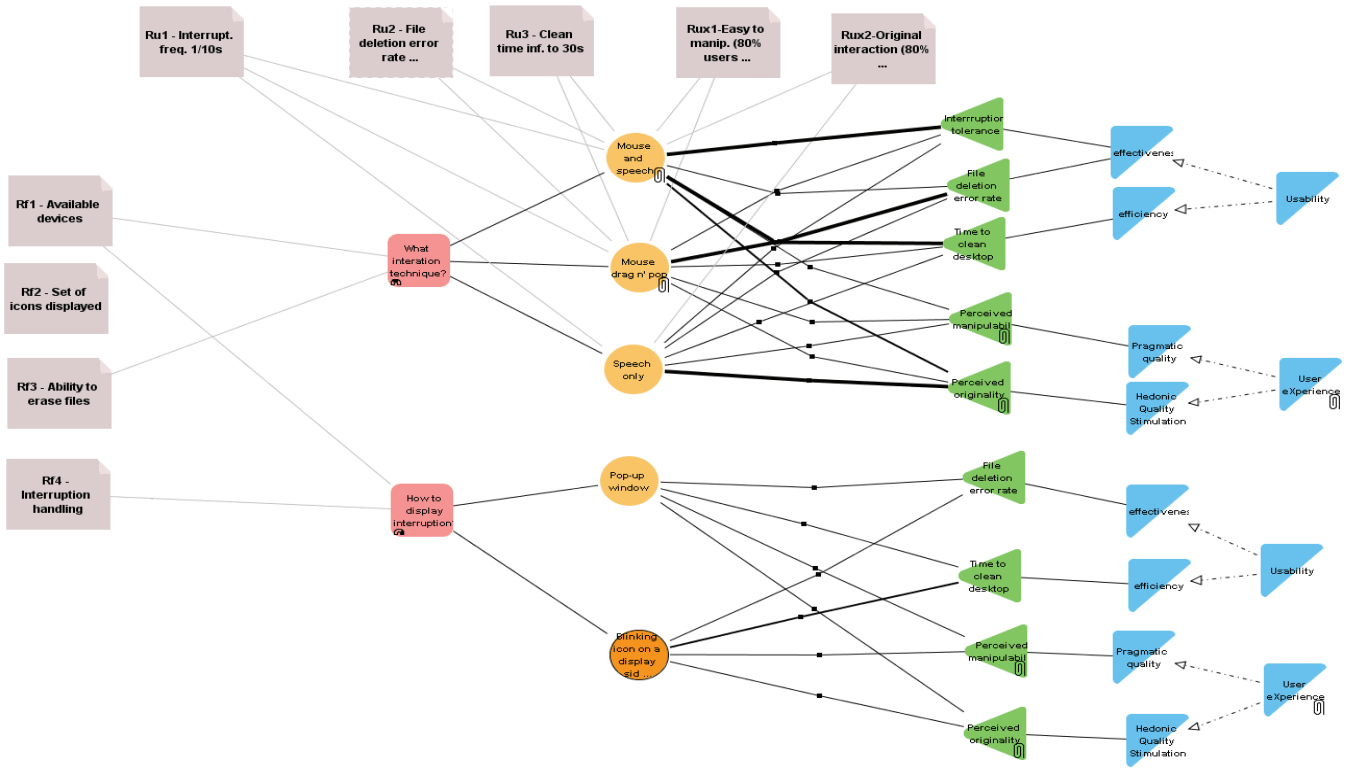

Figure 1. Design Rationale overview from a design session output with DREAM

The right-hand side of the diagram makes explicit the relationship between criteria and factors. Factors correspond to desired characteristics of the system namely "Usability" and "User eXperience". They are in turn decomposed into sub-factors; two out of the three classical ones for usability i.e. "effectiveness" and "efficiency" and two for "User eXperience" i.e. "Pragmatic quality" and "Hedonic Quality Stimulation".

Design options modeling

One of the issues that remain to be solved is how to assess the options with respect to the connected criteria. For instance, how to identify if the option "Mouse and Speech" for the interaction technique is better than the option "Enriched Drag and Drop" with respect to the criterion "Time to clean desktop". In order to address that problem we propose to apply a model-based approach and to define for each option a detailed behavioral description. For that purpose we use the ICO notation [11], but other ones such as [2 or 6] would also be applicable. ICO notation stands for Interactive Cooperative Objects and is a formal notation to describe interactive systems. It is based on object-oriented design pattern and high-level Petri nets. The PetShop associated tool [1 and 13] allows editing the ICO model and prototyping the associated interface at the same time (this aspect is detailed in [12]).

The DREAM diagram shows that a third design option, speech only, had initially been suggested and that this option does not fulfill all the non-functional requirements, even before having prototyped or modeled it. The modeling of an option has a cost indeed and the gain of this approach is effective if the number of modeled and prototyped options is balanced with the modeling and prototyping cost. To help in comparing the two remaining interaction techniques, models of these interactions have been built and embedded in the DREAM diagram as indicated by the paperclip symbol at the bottom-right of each option. ICO models of the two interaction techniques are built and assessed (Enriched Drag and Drop technique ICO model detailed in Figure 2), with respect to both "user experience" and "usability" factors. These models are then verified and prototyped by means of PetShop tool. Effectiveness can be verified and performance evaluation technique can be used to assess time performance for instance. Due to space constraints this is not detailed here. Usability tests can be performed on the prototypes with the building of the application while performance evaluation can be done on the model only. For the user experience aspects, using the prototyped options, an evaluation has been performed with AttrakDiff tool [7] to rate the selected criteria. Paperclip symbols on "perceived manipulability" and "perceived originality" criteria represent the fact that such results are stored in the design rationale model. Furthermore, to help in checking this coverage and to ensure that one or more evaluation criteria are not missing, the tool allows linking the requirements to criteria.

Interpretation and benefits

Once the design options have been assessed, DREAM model makes it easy to perceive which one (if any) has received the best evaluation.

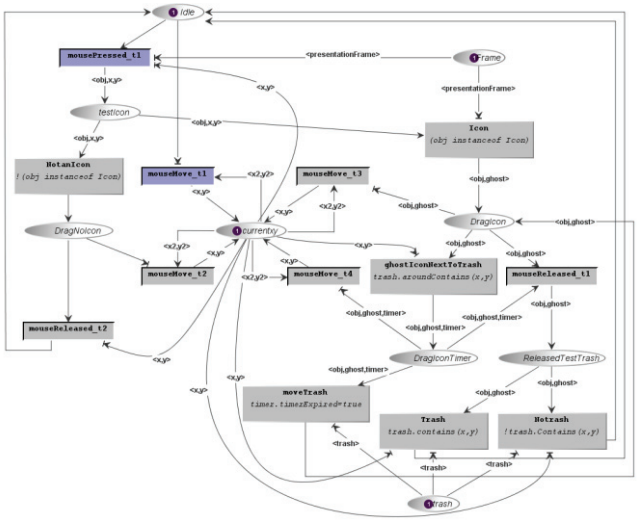

Figure 2. ICO model of Enriched Drag & Drop

In addition it makes it explicit for this option which requirements it fulfils. Coming back to the importance of requirement engineering, DREAM diagram is a critical help to argue about, to select an option as well as to trace back the rationale underlying this selection. In our example, we see at a glance that the "Mouse and Speech" option is the best rated and that it fulfills the entire set of non-functional requirements. The final choice then belongs to the various stakeholders who have additionally direct access to the related requirements which, in turn, cannot be ignored. Furthermore, all the necessary information about the designed interactive system can be gathered and synthesized in one diagram. From a designer perspective, it can help to share conception ideas and materials. From another participant involved in the implementation and/or deployment process, it can be seen as an entry point to the designed system.

CONCLUSION

This position paper argues that various models are useful for the design of interactive systems as they can complement each other and correspond to the needs of different stakeholders. We have presented a model-based approach for description within and single diagram requirements, design questions, design options, criteria and factors. This structured set of information supports different activities such as requirements traceability, design choices decision support and the traceability of design choices decisions. We have also presented how detailed behavioral description of advanced interaction techniques such as "Enriched drag and drop" or "speech and click" can be integrated within that framework to provide additional benefits. Of course such "intensive" model-based approaches require tools to support model construction, analysis, simulation, interpretation and reuse. The CASE tools supporting TEAM and ICO notation are publicly available [3 and 13] and are a corner stone of the applicability of the approach.

REFERENCES

1. Bastide, R., Navarre, D., and Palanque, P. 2002. A model-based tool for interactive prototyping of highly interactive applications. In CHI '02 Extended Abstracts on Human Factors in Computing Systems (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, pp. 516-517.
2. Coninx, K., Cuppens, E., De Boeck, J. & Raymaekers, C., 2007, Integrating Support for Usability Evaluation into High Level Interaction Descriptions with NiMMiT . In: Interactive Systems: Design, Specification, and Verification. 2007, Springr Verlag LNCS, pp. 95-108.
3. DREAM web site: Internet Resource http://ihcs.irit.fr/dream/index.html , accessed Jan 2010.
4. Dutoit, A.H., McCall, R., Mistrík, I., Paech, B., Rationale Management in Software Engineering: Concepts and Techniques, Rationale Management in Software Engineering, p. 1.
5. ESARR 6. EUROCONTROL Safety Regulatory Requirement. Software in ATM Systems. Edition 1.0. http://www.eurocontrol.int/src/public/standard_page/esarr6.html (2003)
6. Hassenzahl, M. The thing and I: understanding the relationship between user and product. In M.Blythe, C. Overbeeke, AF Monk, & PC Wright (Eds.), Funology: From Usability to Enjoy-ment. Dordrecht: Kluwer, 2003, pp. 31-42.
7. Hassenzahl, M. - Internet Resource http://www. attrakdiff.de, accessed Jan 2010.
8. ISO DIS 9241-11 (1996) Ergonomic requirements for office work with visual display terminals (VDT) - Part 11 Guidance on usability.
9. Lacaze, X., Palanque, P., Barboni, E., Bastide, R., Navarre, D., From DREAM to Reality : Specificities of Interactive Systems Development With Respect to Rationale Management, Rationale Management in Software Engineering, pp.155-170.
10. MacLean, Allan; Young, Richard M.; Bellotti, Victoria M. E., and Moran, Thomas P. Questions, Options, and Criteria: Elements of Design Space Analysis. Lawrence Erlbaum Associates; 1991; 6, pp. 201-250.
11. Navarre, D., Palanque, P., Ladry, J., and Barboni, E. 2009. ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. ACM Trans. Comput.-Hum. Interact. 16, 4 (Nov. 2009), pp. 1-56.
12. Palanque P., Ladry J., Navarre D. and Barboni E. High-Fidelity Prototyping of Interactive Systems can be Formal too 13th International Conference on Human-Computer Interaction (HCI International 2009) San Diego, CA, USA.
13. PetShop web site: Internet Resource http://ihcs.irit.fr/petshop, accessed Jan 2010.
14. Rosson, M.B. & Carroll, J.M. 2002. Usability Engineering: Scenario-Based Development of Human-Computer Interaction. San Francisco: Morgan Kaufmann.
15. RTCA. Software Considerations in Airborne Systems and Equipment Certification, DO-178B RTCA, Washington D.C. 1992
16. Sutcliffe A. & Ryan M. Experience with SCRAM, a SCenario Requirements Analysis Method, in Proceedings of the 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice, April 1998, pp. 164-173.
17. Ter Beek, M. H., Faconti G. P., Massink, M., Palanque P., Winckler M. Resilience of interaction techniques to Interrupts: A Formal Model-Based Approach. In proc. of Human-Computer Interaction (INTERACT 2009), Uppsala, Sweeden, Vol. 1, Springer-Verlag, LNCS 5726, p. 494-509.