Toward a Closer Integration of Usability in Software Development: a Study of Usability Inputs in a Model-Driven Engineering Process

Carine Lallemand
Centre de Recherche Public Henri Tudor
29, avenue John F. Kennedy
L-1855 Luxembourg, Luxembourg
carine.lallemand@tudor.lu

ABSTRACT
Even though the benefits of usability have widely been proven, it seems that development-oriented companies face many difficulties to introduce usability practices into their defined development processes. This paper describes the overall methodology deployed as an attempt to achieve a closer integration of usability practices in the software development process. Model-Driven Engineering (MDE) is used as a basis for this integration. Providing a precise framework composed of models and transformations, it allows to track usability problems and to highlight where exactly they occur in the development process. We will thus be able to link every step of the process to specific ergonomic inputs and to study their consequences on the usability of the generated system. Because MDE will only be used as a way among others to investigate some hypotheses on usability and User-Centered Design (UCD) in general, our results are expected to provide valuable and generic information on usability and UCD processes.

Author Keywords
Usability, User-Centered Design, Software Engineering, Model-Driven Engineering, Transformation Rules

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces: User-Centered Design; Evaluation/methodology

General Terms
Design, Human Factors, Measurement, Performance

INTRODUCTION

For more than twenty years, researchers and practitioners have been developing a theoretical background and practical tools in order to improve the usability of systems. It is easily understandable when considering the benefits related to more usable systems. Thus, several studies have shown that usable systems were associated with an increased productivity, reduced errors, reduced need for training and support, improved acceptance and enhanced reputation ([15], [18]).

But even though the benefits of usability have widely been proven, most software developers do not apply correctly any methodology related to usability. The main reasons explaining this fact are related to the time and costs associated with the integration of usability into software development. Following Seffah et al. [21], we also speculate that another reason for this phenomenon could be the lack of reference framework for usability practices that indicates where and how in the software process usability inputs need to be provided [9].

It makes sense, therefore, to try to fill the gap between Software Engineering (SE) and usability practices by studying how we could reach a closer integration of both fields. To address this problem, we decided to focus on Model-Driven Engineering (MDE), a software development methodology, which specifies an automated process for developing interactive applications from high-level models to code generation. The very formal separation of every stage of the development process in MDE constitutes an ideal basis for our study. In this framework, we propose a 3-step methodology, which aims at understanding the relationships between software development stages and usability aspects.

1.1. Usability: a concept with multiple definitions and models

The most common definition of usability is given by the standard ISO 9241-11 [13], which defines it as the "extend to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

Originating from computer sciences, where it was perceived as a human factor in a quality system [24], usability has been also studied in social sciences [6]. Shackel [23] is probably one of the first to define and specify the components of usability that include effectiveness, learnability, flexibility and attitude. However, during the late 1980's and 1990's, several authors focused their attention on the definition of usability, each author relating

EICS'11, June 13–16, 2011, Pisa, Italy.

this concept to different attributes [19]. Usability was also defined in several ways across international standards ([13], [14], [15]). We could also notice that not all authors use the term "usability attribute" to designate the entities, which to them make up the usability. These entities can also be called dimensions, components, scales, criteria or factors of usability [11]. Whatever the term used to identify these dimensions, usability criteria appear to be very numerous and diverse. For the time being, there is no consensus on the definition of usability and its related dimensions. Still, some authors have already tried to build consolidated models ([21], [25]) to go beyond the omissions and contradictions in current models and guidelines.

In the literature, the issue of usability is mostly addressed through the perspective of its evaluation. According to Hornbaek [12], the measures of usability help "make the general and somewhat vague term usability concrete and manageable". However, in his systematic review, the author insists on the fact that a lot of challenges still have to be resolved. The main shortcoming of usability evaluation methods is that they generally need an already developed system or prototype (and most of the time, real users) to produce recommendations ([11], [16]). That will lead to high cost and time and will not allow intervening at the earliest stage of the development process. Moreover, most of the usability criteria are difficult to translate into accurate metrics that can be implemented into the code. Therefore, it is hard for the developers to integrate usability at the early stage of the development process.

Finally, these findings lead us to several conclusions. On the one hand, it is obvious that the overarching conceptual framework of usability lacks consensus and there seem to be no models that deal with all the requirements. This lack of a consistent and consolidated framework for usability is probably one of the main reasons why usability is still not well integrated into SE practices. On the other hand, even if the tools provided by the field of usability are able to assess a system's degree of usability, they are still not really able to provide predictive guidelines which will ensure usability at the design level or even just guide developers by classifying the usability criteria into the different stages of their defined processes. Our project aims at investigating and, so far as possible, solving these issues. We will therefore look at how some studies attempt to integrate usability principles or practices in the software development process.

1.2. Integration of Usability in the Development Process

In the 1980's and 1990's, usability was seen by developers as an issue only related to the presentation of information, before they included system functionality in their concerns. As stated by Seffah et al. [22], this narrow view of usability was definitely not able to ensure the whole usability of a system. Fortunately, further works came to the conclusion that even non visible system features would impact the interaction between the user and the system [4]. Starting from this observation, researchers and practitioners have been concerned with the goal of achieving usability through software architecture ([3], [11]). Some authors provide architecture mechanisms or design patterns that directly relate to usability aspects ([3], [22]). Other studies try to define strategies to introduce HCI techniques and activities into mainstream software engineering practices [10]. In the same manner, the famous international standard ISO 13407 [15] proposes a framework for the integration of usability at all stages of a development process. Its wide concept of User-Centered Design (UCD) is described as an "approach to interactive system development that focuses specifically on making systems usable".

Even so, it seems that current usability engineering practices fail to drive design at all stages [11]. We make the assumption that some concepts or methods introduced by SE could be useful to answer the problems emerging from the field of usability and to reconcile closely SE and usability. We assume that Model-Driven Engineering could constitute a favorable framework to our project.

1.3. Model-Driven Engineering: a Way to Achieve Closer Integration of Usability into Software Development?

Model-Driven Engineering (MDE), and its variant Model-driven Architecture, have recently attracted the interest of both researchers and practitioners and are currently seen as key perspectives in the field of SE. MDE is a software development methodology which specifies an automated process for developing interactive applications, based initially on models of abstract description of the system, manually specified by the software developer (user model, task tree, etc.). These models are gradually transformed into transient models of description, more concrete, of the interactive system (Abstract User Interface and Concrete User Interface), resulting in the generation of executable source code, the basis of the Final User Interface. The Cameleon Reference Framework [5] is one of the most commonly used tools for MDE.

By separating the design tasks from the development ones and applying progressive model-to-model transformations, MDE allows the usability to be integrated into the whole user-system interaction process and not only at the graphical level of the user interface. As Juristo, Moreno and Sanchez-Segura [17] insist, user interface is only the visible part of the system but "interaction is a wider concept".

Some authors are already conducting studies linking MDE and usability ([1], [2], [7]). Abrahão, Iborra and Vanderdonckt [1] try to show how the usability of user interfaces that are generated by an MDA-compliant tool can be assessed. They introduce the idea of "usability by construction" and already imagine future trends: "It is our belief that model-driven development provides the basis for tight integration of usability evaluation in the MDA development process, allowing usability issues to be addressed as an integrated part of the system design and not just as an ad-hoc solution after most of the development has been completed". Our research goal is closely related to this

work since the authors are investigating whether MDE methods improve software usability through model transformations. Fernandez, Insfran and Abrahão [8] provide a usability model to evaluate usability at several stages of a MDE-compliant development process. We could possibly adapt or extend this model to reach our research objectives and test our hypotheses.

Finally, it seems that MDE conveys new perspectives for the usability research field. It allows to track usability problems and to highlight where exactly they occur in the development process. The fact that it constitutes a precise framework composed of models and transformations, allows us to link every step of the process to specific ergonomic inputs and to study their consequences on the usability of the generated system, at a global or specific level. Thanks to these properties, MDE constitutes an ideal basis to reach our objectives. However, it emphasizes once again the need for operationalization of ergonomic rules. Thus, further research is needed to achieve true integration of usability into software development. We aim at going further than the existing studies by providing a whole integration of UCD into MDE. We will therefore focus on the transformational approach in MDE, as this methodology provides an ideal basis for the integration of usability at every stage of the development process.

2. RESEARCH OBJECTIVES

The overarching research question is formulated as follows: How and to what extent is it possible to provide a solid framework for User-Centered Design in order to reach a closer integration of usability in software development?

In terms of scientific contribution, our project will try to counter the limitations of the current usability models and guidelines. We will use MDE, a SE tool, in order to deepen the understanding of how usability inputs should be implemented within the development process. Because MDE will only be used as a way among others to investigate some hypotheses on usability and UCD in general, our results are expected to provide valuable and generic information on usability and UCD processes.

The research questions addressed by this study are numerous: (i) Based on the existing usability models, what usability model could be the most appropriate to improve the integration of usability in the software development lifecycle? (ii) How to structure the ergonomics inputs according to the different steps of a design process in order to build a shared framework for both usability specialists and software engineers? (iii) By studying the usability inputs into the software architecture, could it be possible to evaluate the usability of a system, based on the coverage (in number and quality) of each specific usability dimensions at the different stages of software development?

3. METHODOLOGY

Our methodology will be structured in 3 major steps.

In a first step, a systematic review of the literature on the usability concept will be performed to investigate the different definitions of usability and its related dimensions. Based on the existing usability models, we aim to find, adapt or define a consolidated usability model that would be appropriate to improve the integration of usability in the design process. The criteria for the definition of this model will encompass: the quality and orthogonality of the usability dimensions, the ability to be used for the design as well as for the evaluation of a system and the ability to be operationalized in a development process.

The second step aims at classifying the various dimensions of usability defined in step 1 according to the different stages of an MDE development process. At this stage, our purpose is to know where the different usability dimensions would play a role and, consequently, where they would have to be taken into account and implemented. We aim to create a table that establishes a classification of the usability dimensions identified in our consolidated model and the different stages of the development process.

The last step of the methodology encompasses: a stage of use cases design and a stage of usability analysis performed on these use cases. We plan to use at least 2 use cases. In collaboration with IT-engineers, we will develop three different interfaces for each use case. Following the Cameleon Reference Framework [5], our MDE design approach will cover 4 steps that consist of applying transformations rules in order to move progressively from Task and Domain Models to a Final User Interface. The difference between these 3 interfaces will lie in the operationalization of usability criteria into transformation rules. The number of transformation rules covering each criteria and the quality of the coverage will therefore vary and we will thus be able to compare the usability of each generated system. The usability of the generated systems will be assessed through 3 usability analyses: a heuristic evaluation performed on the transformation rules during the development, a heuristic evaluation performed on the final user interface and, finally, users tests performed in a usability laboratory.

In summary, the results obtained through this methodology will allow us first to evaluate the quality of the generated interfaces. Second, we will be able to investigate the possible links between the coverage of each specific usability dimension at the different stages of software development and the whole usability of the final interfaces. We also aim to formulate a set of propositions for the improvement of user-centered design and the integration of usability into the development lifecycle, both as a basis for further research work and for the attention of practitioners.

CONCLUSION

In this study, Model-Driven Engineering is used as a basis to achieve a closer integration of usability at the different stages of software development. The very formal aspect of this approach will allow us to take support on a precise and well-defined process to track usability problems at the different level of abstraction, which could represent the

different levels of a development lifecycle. If possible, we would aim at generalizing our results to every development lifecycle, this way giving the practitioners a better understanding of the usability related issues. Usability specialists could derive benefits from our results, especially by being more integrated in the earliest stages of the development process. We also aim at providing them with relevant information on the link between transformation rules and usability dimensions.

ACKNOWLEDGMENTS

The author would like to thank her corporate and academic supervisors: Dr. Guillaume Gronier (CRP Henri Tudor), Dr. Vincent Koenig and Pr. Romain Martin (University of Luxembourg). She would like also to thank Pr. Jean Vanderdonckt for his valuable advices and support.

REFERENCES

1. Abrahão, S., Iborra, E., Vanderdonckt, J. Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool. In Law, E. & al. (Eds) *Maturing Usability: Quality in Software, Interaction and Value*, Springer (2008), 3-32
2. Aquino, N., Vanderdonckt, J., Condori-Fernández, N., Dieste, O., & Pastor, O. Usability evaluation of multi-device/platform user interfaces generated by model-driven engineering. *In Proc. ESEM'2010*, ACM Press (2010).
3. Bass, L, & John, B. E. Linking usability to software architecture patterns through general scenarios. *Journal of Systems and Software*, 66, 3 (2003), 187-197.
4. Bass, L., John, B. E., & Kates, J. *Achieving Usability through Software Architecture*, Carnegie Mellon University/Software Engineering Institute Technical Report No. CMU/SEI-TR-2001-005, 2001.
5. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15, 3 (2003), 289-308.
6. Carroll, J. M. Human-computer interaction: psychology as a science of design. *Annual review of psychology*, 48 (1997), 61-83.
7. Fernandez, A., Abrahão, S. & Insfran, E. Towards to the Validation of a Usability Evaluation Method for Model-Driven Web Development, *In Proc. ESEM 2010*, ACM Press (2010).
8. Fernandez, A., Insfran, E., & Abrahão, S. Integrating a Usability Model into Model-Driven Web Development Processes. *In Proc WISE 2009*, Springer (2009)
9. Ferre, X., Juristo, N., & Moreno, A.M. Framework for integrating usability practices into the software process. *In Proc. PROFES'05,* Springer (2005).
10. Ferre, X., Juristo, N., & Moreno, A.M. Improving software engineering practice with HCI aspects. Software Engineering Research and Applications, Springer-Verlag, San Francisco, USA (2003), 349-363
11. Folmer, E., & Bosch, J. Architecting for usability: a survey. *Journal of Systems and Software*, 70, 1-2 (2004), 61-78.
12. Hornbaek, K. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64, 2 (2006), 79-102.
13. International Organization for Standardization. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 9: Guidance on usability, 1998.
14. International Organization for Standardization. ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model, 2001.
15. International Organization for Standardization. ISO 13407: Human-centred design processes for interactive systems, 1999.
16. John, B., & Marks, S. Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology*, 16, 4 (1997), 188-202.
17. Juristo, N., Moreno, A.M. & Sanchez-Segura, M. Analysing the impact of usability on software design. *The Journal of Systems and Software*, 80 (2007), 1506-1516
18. Maguire, M. Methods to support human-centred design. *International Journal of Human-Computer Studies*, 55, 4 (2001), 587-634.
19. Nielsen, J. *Usability Engineering*. San Diego, CA: Academic Press, 1993.
20. Scapin, D., & Bastien, J. M. C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, 16, 4 (1997), 220-231.
21. Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. Usability measurement and metrics: A consolidated model. *Software Quality Journal*, 14, 2 (2006), 159-178.
22. Seffah, A., Mohamed, T., Habieb-Mammar, H., & Abran, A. Reconciling usability and interactive system architecture using patterns. *Journal of Systems and Software*, 81, 11 (2008), 1845-1852.
23. Shackel, B. Usability - context, framework, design and evaluation. In B. Shackel & S. Richardson (Eds.*), Human factors for informatics usability*. Cambridge: Cambridge University Press, 1991, 21-38.
24. Shneiderman, B. Improving the human factors aspect of database interactions. ACM Transactions on Database Systems, 3, 4 (1978), 417-439
25. Winter, S., Wagner, S., & Deissenboeck, F. A Comprehensive Model of Usability. *In Proc. EIS 2007*.