Work in Progress - Introducing Usability Concepts in Early Phases of Software Development

Yonglei Tao
School of Computing and Information Systems, Grand Valley State University, Allendale, MI 49401 taoy@gvsu.edu

Abstract - Usability is critical for the success of interactive software systems. However, usability evaluation is rarely taught as an indispensable element of the development process in software design courses. On analyzing usability errors students made in their programming projects, we found some usability errors, although revealed via the visual design of a user interface, resulting from earlier design decisions. It is beneficial to introduce usability concepts in early phases of software development. In this paper, we present an approach to integrating usability evaluation into behavioral modeling for interactive systems. Early usability evaluation is intended to help students obtain a better understanding on usability and encourage them to make conscious effort in seeking usability throughout the development process.


INTRODUCTION

Usability is a key quality attribute for the success of interactive software systems. Due to a perceivable gap between Human-Computer Interaction (HCI) and Computer Science (CS), however, usability evaluation is rarely taught as an indispensable element of the software development process in software design courses [2][3]. As a consequence, HCI is considered by software professionals as one of the most important knowledge areas that are the least taught in their formal education and therefore require the greatest on-the-job learning [1].

Usability evaluation is usually performed towards the end of a software project, that is, after the design decisions and implementation details on the user interface are fully determined. But certain usability errors occur early in the development process. If unnoticed, they may propagate to the successful phases and potentially affect many aspects of the final product [4]. Even though an iterative development process is promoted, students tend to cut short their development effort to meet the project deadline, leaving many usability errors intact.

As evaluating a relatively complex programming project we assigned to our students in a software design class in the past three semesters, we found 27% of usability errors that students made or 38% of points they lost resulting from an inadequate design on the interaction part of the system, rather than the visible part of the user interface. Here, points lost are intended to show the weights of those errors. Some usability errors, although revealed via the visual design of a user interface, were conceived much earlier. It suggests that usability be taken into consideration not just after visual design, but also in early development phases.

Early usability evaluation is beneficial not only for developing a usable software product but also for instructional purposes. It makes possible to introduce usability concepts in early phases of software development, helping students obtain a better understanding on usability and encouraging them to make conscious effort in seeking usability throughout the development process.

OUR APPROACH

Understanding user-system interaction is essential for developing an interactive system. At a higher level of abstraction, user-system interaction is essentially about how the user navigates screen to screen as well as through interaction modes within each screen in order to accomplish one’s objectives. Users are able to use an interactive system without much problem if they know where they currently are in the system, where they have been, and how to get where they want to go [7].

A state machine model is often used in software design to describe system behavior. For an interactive system, such a model also captures user-system interaction as well as the context in which the interaction takes place [5]. We use statechart diagrams [6] to define such a model and refer to it as the behavioral model.

In addition to basic notations such as states, state transitions, events, and actions, statechart diagrams provide composite states, that is, states containing substates. In the behavioral model, a composite state represents a screen and a substate an interaction mode. We stress that events must be externally observable, resulting from user and computer interacting via visual components that make up the user interface, such as the user clicking on a command button and the system bringing up a message box. And a sequence of events describes steps that the user takes for accomplishing a task.

As we observed, students did not have much problem with understanding the behavioral model due to a previous knowledge about state machines. But they did not know how to handle certain design decisions. Especially, when there were several options for a particular issue, they did not know on what basis to make an appropriate decision.
A simple example is to choose the default mode for a screen that allows the user to add, edit, and delete a record. At least there are two options. One option is to introduce a mode in which the user can determine what to do next. And the other is to make the default a mode in which the user can immediately work on one of the basic tasks. Apparently, commonly used evaluation criteria in software engineering, such as correctness, consistency, and completeness, are not of much help in this case. Instead, such a decision must be made according to the usage profiles and frequencies of tasks which are part of the usability requirements. If several tasks are equally likely to perform at the beginning or it is important to meet the need of certain users (such as inexperienced users), the former is more preferable. Otherwise, if the user starts the application in most cases with performing a particular task, the later can improve the overall usability.

Usability is the capability for an interactive system to be used by humans easily and effectively [5]. As shown above, usability is an indispensable evaluation criterion for the behavioral model. However, students won’t be able to do it without appropriate guidelines.

Various usability evaluation techniques are available [5]. Nielsen’s experience shows that heuristic evaluation is especially suitable for instructional purposes [8]. Heuristic evaluation is to analyze a user interface for its compliance with established usability principles, i.e., the heuristics [9].

Nielsen proposes ten usability principles for heuristic evaluation. In fact, several of the principles do not entirely reflect on the visible part of a user interface; they also address the flow of interaction. In other words, these principles can be applied to the behavioral model, even though the visible part is not available yet. We identified such principles and explained them to students in relation to common usability problems. Explicit tie-in between general principles and specific usability problems makes the otherwise abstract usability concepts concrete.

Moreover, we developed a non-trivial problem as a class exercise and organized students’ activities around a design-evaluate-redesign cycle. Students were able to discover certain usability errors which would pass unnoticed without using the selected principles as guidance.

**SUMMARY**

Various proposals for early usability evaluation can be found in the literature [5]. One of the widely known is GOMS, a family of user interface design and evaluation tools [10]. Except for their theoretical values in a HCI course, those techniques are mostly too complex to use for instructional purposes.

Our approach is to applying usability evaluation to the behavioral model. It does not require much additional effort. A state machine model is not new to our students and heuristic evaluation is extremely flexible and time-efficient. As such students can focus on design issues rather than struggling with new notations.

In our approach, the behavioral model is a mechanism for students to visualize their perception about the user-system interaction. It allowed students to apply the selected usability principles to discover errors and improve their user interface design. It also helped us to understand the nature of some common errors on students’ projects, which was not possible otherwise.

From the instructional perspective, it is beneficial to separate design on the interaction part of a user interface from its visual part. Students can focus on certain usability issues in early phases and then look into others as they become relevant in the subsequent phases. Usability considerations are multi-dimensional. Progressive learning is an effective way for students to gain a thorough understanding on usability concepts.

One comment several students made after working out the above-mentioned class exercise is that “the problem was more complex than it appeared to be at first”. Certainly the problem remained the same. What was different was that students were able to take an additional criterion, that is, usability, into consideration, resulting in an enriched learning experience.

Due to its limitation, our approach is insufficient by itself to accomplish all the goals of usability evaluation. While continuing our research on integrating usability evaluation into early phases of software development, we need to expand our approach so as to provide students with a more comprehensive view about usability.

**REFERENCES**


