

Student centered HCI teaching

Marta Kristín Lárusdóttir
School of Computer Science
Reykjavik University
Ofanleiti 2
103 Reykjavik, Iceland
marta@ru.is

WORKSHOP THEME

After teaching about 500 students the fundamentals of human-computer interaction on their first semester in Computer Science, I have seen that the design process is the most difficult part for all my students. In their HCI course they go through the four phases of software development: analysis, design, programming and evaluation. The goal of the course is to give students insight into the many methods and models that are applied in HCI today and at the same time, give them a good practical experience in using one method for each phase.

After this experience I am eager to conduct a forum for HCI teachers to share opinions on what the biggest problems for the students in learning HCI are, in what phase of the software development do they have the most difficulties, why do the students experience these troubles, and what have other teachers done to help the students overcome these.

WORKSHOP BACKGROUND

During the HCI course at Reykjavik University students go through all four phases of software development: analysis, design, programming and evaluation. Additionally, they are introduced to the fundamentals of HCI and the issues of writing help text. The course is based on the book: Software for Use, by Constantine and Lockwood [1]. The students attend 3 lectures and 2 discussion/problem classes each week for 12 weeks. In all, about 500 students have taken this course now. In the following I will share my experience of teaching this course.

User and task analysis

In this phase the students are introduced to usability and the need to analyze users, their tasks and the environment users are working in, to be able to make usable systems. Personas and scenarios are introduced, whereas user roles and essential use cases are taught thoroughly. They do exercises in describing user roles and essential use cases both in discussion classes and as home assignments.

Describing the user roles and essential use cases has been fairly easy for the users. Some of them find it hard to find the line between being not too general and not too detailed, when making the use case descriptions but they all seem to find a rhythm after a few iterations.

Abstract prototyping

After the analyzing users and tasks for two weeks, the students concentrate on learning to design the user interface for two weeks. First they learn how to do abstract prototypes or the content model, as it is called in [1]. There the emphasis is on finding the tools and the materials in the user interface that are needed to solve the users tasks by writing each element on a post-it note. Additionally the tools and materials are grouped in information spaces, which will become a window, a tab or a panel in the final design.

Here the trouble begins. It is very hard for the students to understand abstract prototyping. They are kind of shy using post-it notes to design. Maybe the reason is partly that some of the students have not programmed much so they don't grasp at once what an input field is and partly it is because they don't really see the purpose of doing abstract prototypes. They want to do more detailed prototypes or even start programming!

Additionally some of them are thinking very procedural. They make one post-it note for each step in the use case and lay out the notes in the information spaces in the same order as the steps in the use case description beginning in the top-left corner and going down to the lower right corner, so they really want to grasp in what sequence the user would do things.

After doing the abstract prototypes the students do a detail prototype in Excel, providing some navigation between windows.

The process of design is remarkably hard for the students to understand. The idea of doing the user interface design in two steps is very good, but the content model idea needs to be improved or maybe given a more detailed description for the students to be able to use it in a better way.

Navigation map

After doing a detailed design of the user interface the students make a navigation map.

The navigation map has also been a great barrier for the students. Some of them do not understand the difference between a window and a panel on the window. Many of them also complain that the instructions are not good enough. After going through all the books I have on HCI only one included some text on navigation maps. This is quite surprising. Doing navigation maps is a very good tool to get an overview of the system both for developers and users, and also to check out if the user interface is too complex or not. Drawing tabs in the navigation map has been especially hard for the students, so giving an example of that would strengthen the material a lot.

Programming

Next, the students program the case they have been analyzing and designing in home assignments. This is the first time they use C-Builder for programming, and the focus changes suddenly to all these technical matters of programming and is NOT on the usability. A course in HCI might be better placed on the second semester, because sometimes it is like teaching them the alphabet and how to read out-loud in a listenable way at the same time. They need some experience in programming by doing one complete software program, before adding the quality matter, usability, to the picture.

Evaluation

Students are introduced to many evaluation methods, both inspection methods and usability testing methods, and they get practical experience in using both Heuristic evaluation and the Think-a-loud method in home assignments. For Heuristic evaluation they use the rules and principles in the book for the course: *Software for Use*, chapter 3 [1], as a basis for their evaluation. They do the Think-a-loud test both on their own project and on a Web-solution they have not been involved in doing. Teaching and learning the evaluation seem to be fairly straight forward, all the students are very impressed how much benefit they get from observing the users in a usability test, especially on their own projects, so this has been a great experience both for the students and the teacher!

Conclusion

After teaching about 500 students the fundamentals of human-computer interaction, how to analyze user roles, do

essential use case descriptions, abstract prototyping by doing a content model, navigational maps, program their design and evaluate the outcome, I've come to the conclusion that the design process of the user interface is the most difficult part for the students. Better material and more research are needed to improve the teaching of these subjects. Additionally the students need some experience in programming and the software development live-circle before adding the quality matter, usability to the picture.

WORKSHOP AIM

The aim of the workshop is to create opportunity for teachers to share experiences on what are the biggest problems for their students learning HCI and get an inspiration from another on how to solve these problems. The following questions will be discussed:

- a) In what phase do the students experience the most difficulties?
- b) Why do they experience difficulties? Are the methods or the set of techniques we are using not good enough? Or is it the teaching material that does not communicate to the students?
- c) What solutions are there to help them through these?

WORKSHOP PROGRAMME AND OUTCOME

Introduction: The organizers introduce the workshop aim and the participants introduce themselves

Discussions: The participants will split into groups for discussing the questions above. Each group will try to reach a common understanding.

Outcome: The groups will present the results of their discussions. These will be collected and sent to all the participants after the workshop.

PARTICIPANTS AND ADMITTANCE

This workshop is targeted at experienced teachers in HCI that want to share experience with each other on what difficulties their students have.

WORKSHOP ORGANIZER

Marta Kristín Lárusdóttir, is an assistant professor in the Computer Science Department of the Reykjavik University teaching and researching HCI. Her research work focuses on evaluation both in the field and in the laboratory. Marta has been taking part in two European research projects, published papers and given talks at conferences.

REFERENCE

1. Constantine, L. L. and Lockwood, L. A. D. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Reading, MA: Addison-Wesley, 1999.