Teaching Human Computer Interaction to Programmers

Saul Greenberg

Abstract. Many computer science graduates will likely find themselves developing interfaces in a work culture that has only naïve notions of usability engineering. This article describes a course I developed that prepares students for this eventuality by providing them with practical and applicable HCI skills. All course material is available on the world wide web, and pointers are provided.

The Challenge.

A recent phenomena in the computer industry is the expectation that everyday programmers, such as those working in small firms producing in-house software, will design good interfaces as well as good code. Unfortunately most programmers are sadly unprepared for this job. Their traditional computer science training rarely included HCI, either because courses were unavailable in their educational program, or because such a course was considered esoteric and for specialists.

This, of course, is changing. Because of job demands, many computer science students now consider HCI a core skill as marketable as (say) databases and networking, and HCI courses are becoming well attended. At the University of Calgary, for example, the department has offered an undergraduate HCI course since 1981, but it is only recently that it has grown from a 'specialist' course with 30-40 students, to a heavily attended mainstream course with 100 students (i.e., about three-quarters of all computer science majors).

The question that I face as an educator is how to fashion computer scientists into HCI practitioners with appropriate background and skills. Because Alberta has a large oil and gas industry with fairly traditional data processing departments, I expect most students will work in groups where the term "HCI" is unknown, or at best that their managers would have fairly naive notions of what "good" interface design is all about (e.g., that interface design is knowing how to program Visual Basic). I would not only have to teach students fundamental HCI principles and foundations, but would have to give them skills that they could use in a work environment unfamiliar with the idea of usability engineering (Nielsen, 1993).

While the ACM SIGCHI Curriculum is an obvious source and inspiration to HCI educators (Hewett et. al., 1992), I found that I could not use it directly. The document is at its best when considering how HCI can be integrated in a curriculum that would produce HCI specialists. In our department, HCI is a single course, and it is unlikely that a curriculum redesign would be well received by other faculty.

The Course.

The course that I offer to students is neither perfect nor complete, and I do not expect all educators to agree with my approach. However, many of my students do seem to become reasonably adept at applying their learning to practical situations.

Students. Students are typically undergraduates pursuing a computer science major at the University of Calgary, and are usually in the third or last year of the degree program. They already have basic computer science skills (programming, data structures, software engineering), but only a few will have taken introductory psychology or statistics courses as an options. Most take the course because they see it as a marketable skill, and few would say they are pursuing a career as an interface designer.

Purpose. The course presents HCI as a usability engineering process that integrates the design, implementation and evaluation of interfaces (Figure 1). The bottom line is that students should have sufficient skills to design, implement, and evaluate reasonable interfaces in real life work environments, even when they may not have a good budget or time allowance or managerial support to do so.

On completion of the course, students will understand what is meant by good design, and will have experienced designing systems that are usable by people. Students will know contemporary techniques for implementing interfaces, and will have applied these to building applications through paper prototypes and graphical user interface toolkits.

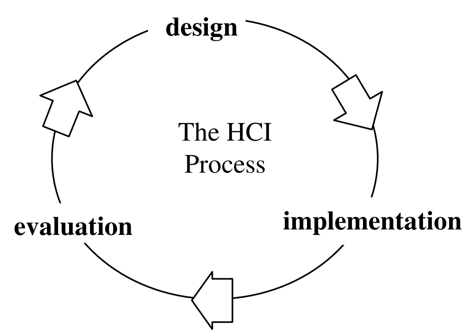

Figure 1. HCI as a usability engineering process

Sections	Topics
1 Overview	• Introduction to the course & to HCI
2 Understanding Design	• Psychology of everyday things • History of HCI
3 Designing with the user	• Evaluating interfaces with users • Involving users in the design process
4 Designing without the user	• Task-centered system design • Models of human behaviour • Usability principles & heuristics • Creativity and metaphors • Graphical screen design
5 Implementing GUIs	• Windowing systems and toolkits • The Tcl/Tk language
6 The future	• Visions of the future

Table 1. Major topics in the course.

Students will also know and have practiced a variety of simple methods for evaluating the quality of an interface.

Structure. The course unfolds by examining design, implementation, and evaluation as a continual, integrated, and iterative process. Theoretical class lectures are augmented by case studies of interface successes and failures; students are expected to provide examples of problems they have had with computers and contribute to class discussion. Students will also apply the theoretical knowledge learnt to series of assignments that brings them through an entire design, implementation, and evaluation cycle. The outline of major topics is presented in Table 1.

Course text. The course text is the recent book *Readings in Human Computer Interaction: Towards the Year 2000* (2nd Edition), by R. Baecker, J. Grudin, W. Buxton, and S. Greenberg. Aside from being one of its authors, I chose this book because it contains a huge amount of material related to HCI. Each of its fourteen chapters introduces and briefly surveys a fundamental topic in HCI, includes important papers written by original authors, and includes many pointers to other literature and technical video sources. Because of its richness, I can design the course around the book, rather than have the book force me down a particular curriculum path. I also feel that its breadth and depth makes it an excellent source for students for either in-depth reading onto a particular topic, or to topics that are outside the immediate course requirements.

Assignments. In the three course assignments, students practice and apply particular methodologies they have learnt in class. They pursue a controlled experiment and quantitative evaluation in Assignment 1, a usability study and qualitative evaluation in Assignment 2, and a major project on interface design and implementation in Assignment 3. The assignments are worth 50% of their total grade, with examinations covering the other half.

Availability. Almost all course material is available through the World Wide Web (Table 2). Topic descriptions are formatted as html pages, with links to postscript files containing overheads, assignments, and handouts. The URL is indicated in the table.

References

1. Baecker, R., Grudin, J., Buxton, W., and Greenberg, S. (1995) *Readings in Human Computer Interaction: Towards the Year 2000* (2nd Edition), 950 pages, Morgan Kaufmann Publishers, California. ISBN 1-55860-246-1.

2. Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., and Verplank, W. (1992) *ACM SIGCHI Curricula for Human-Computer Interaction,* Report of the ACM SIGCHI Curriculum Development Group, ACM.

3. Nielsen J. (1993). *Usability Engineering*, Academic Press.

Pages formatted as html	*Pages formatted as postscript*
• descriptions of each topic	• all overheads (as postscript files)
• associated readings from the text	• copies of all handouts
• relevant videos that I show in class	• details on all assignments
• major sources I use to prepare lecture material	• notes for teaching assistants.
• in-class teaching tips	
• overheads are now being translated into html	

Table 2. All course materials, structured as html and postscript pages, are available through the World Wide Web:
http://www.cpsc.ucalgary.ca/projects/grouplab/481/481.html