Teaching HCI across the curriculum

Claudia Roda (croda@ac.aup.fr)
The American University of Paris

This position paper reports on some reflections and open questions arising from teaching an undergraduate level HCI course at the American University of Paris (AUP) to a group of students majoring in Computer Science or International Communication. Although I have worked for many years in the field of agent based systems and I have taught many computer science courses, I am new to the teaching of HCI. I have run my first HCI course this semester and I have had quite a few very pleasant surprises especially in relation to the work I am doing within a committee charged to review the math, science and technology general requirements for the University (general requirements are all those requirements that should be met by all graduating students notwithstanding their major).

Because HCI sits at the cross-roads of so many different disciplines, I would like to discuss with other workshops participants their feelings and experiences in using HCI courses as the contact point for University departments that generally do not interact enough (or at all). The HCI community has often focused on the interdisciplinary characteristics of HCI degrees seeing HCI as a discipline toward which many departments should contribute. Here, I would like to focus on the opposite perspective, looking at the contribution of introductory HCI courses to the curriculum of many departments introducing students to disciplines different than their major and to digital technology.

Basically HCI introductory courses could respond to the two following needs:

1) **The need for interdisciplinary studies:** Students graduating from higher education institutions are increasingly likely to both work in interdisciplinary environments and to have a major shift of career requiring them to learn a new discipline. As a consequence Universities should give students the chance to learn and experiment with contents and methodologies which are not immediately related to their disciplines and acquaint them with the critical reasoning necessary to understand the relationships existing amongst different disciplines. These possibilities are often not offered in European Universities. The major/minor structure of British/American Universities allows students to learn different subjects but normally no effort is made to encourage students to discover relationships amongst different disciplines.
2) **The need for an introduction to digital tools across the curriculum:** (1) what these tools should look like, (2) what one should expect from them, (3) how they can be used efficiently, (4) when they should be accepted, rejected or modified, and (5) how to evaluate the impact of the tools on one's working habits or lifestyle. This knowledge should be supplied across the curriculum; it is in fact as essential for the designers of digital tools as it is for the users.

In my experience teaching HCI to a class composed of students majoring in Visual or Corporate Communication, Journalism and Computer Science, I was pleasantly surprised by the immediate involvement of all the students in the various aspects of HCI.

<u>Students learned about disciplines different than their own</u> - Students with a technical background had the chance to discover the humanities and social sciences. For instance they were introduced to the contents and methodologies of sociology and management by studying organisational impact of software design. They discovered rhetoric by studying the efficient use of language in the interfaces, user manuals, error messages, etc. They became acquainted with visual communication by studying the graphical aspects of interface design. On the other hand, non-computer science students were exposed to the contents and methodologies of Computer Science and they had the chance to see all the life cycle of software products and even learned how to write simple scripts. Students with different backgrounds were able to help and

challenge each other as well as maintaining their discussions lively and interesting during all parts of the course. Often we had to discuss the different goals underlying the design of software products at the levels of marketing, distribution, design, etc.

The course worked as a good introduction to technology - Most courses introducing digital technology focus on a set of simple software products (word processors, spreadsheets, and presentation software). Whilst these courses are useful because they supply the students with the basic skills needed for research and work, they normally do not provide a more general understanding of the differences between the characteristics of a specific brand product and the more general concepts associated to a whole category of software tools. The study of user centred software design principles allows students from all backgrounds, to clearly see how the same core task can be described and presented to the user in many different ways. It shows how design choices imply a conceptual model and how this conceptual model may be or not close to the user's conceptual model. Students learn the need to consider personal and organisational impact. They learn to evaluate the usefulness of the support material ranging from user's manuals to error messages. They basically gain the meta-knowledge necessary for the evaluation of software products in general rather than concentrating on specific ones.

The contents of my introductory course:

1) Overview of HF/HCI Chapter 1, Shneiderman: Designing the user interface, Addison Wesley
2) Societal impact of technology Afterword, Shneiderman (ibid.)
3) Software design "Bringing design to software", by Terry Winograd, Addison Wesley 1996: Introduction "A software design manifesto", by M.Kapor 1991reprinted as Chapter 1 of Winograd (ibid.)
4) A brief history of HCI "A Brief History of Human Computer Interaction Technology." B. A. Myers, ACM interactions. Vol. 5, no. 2, March, 1998 pp. 44-54
5) Interface agents Papers from "Software Agents", M. Bradshaw editor, AAAI Press /MIT press 1997:
   a) "Agents: from direct manipulation to delegation" by N. Negroponte
   b) "Interface agents: metaphors with character" by B. Laurel
   c) "Designing agents as if people mattered" by T. Erickson
   d) Online: Letizia Demo and related papers
6) Theories, principles and guidelines Chapter 2, Shneiderman (ibid.). Emphasis on Object-action interface model
7) Conceptual model "Design of the Conceptual Model", Chapter 2 of Winograd (ibid.)
8) WWW as a system Chapters 2 and 16, Shneiderman (ibid.)
   a) Emphasis on object-action interface model for Web design
   b) Client side scripting
9) Managing the design processes Chapter 3, Shneiderman (ibid.)
10) Assessment methodologies (expert reviews, usability testing, etc.) Chapter 4, Shneiderman (ibid.).
11) Presentation styles Chapter 11, Shneiderman (ibid.).

Ongoing activities: (1) Review of assignments (comparison of interface, conceptual models, functionality of various software products) (2) Discussion of some of the latest Jacob Nielsen columns and HCI related articles (3) Review of final project and meetings with final project's "customer".

Open questions

1. Would HCI courses be better taught by a team of teachers from several backgrounds (computer science, business / management, communication / rhetoric / graphics, sociology? If yes has anybody had the experience of co-teaching a course this way? How did they organise the course? What were the major challenges?
2. Having a class composed by students with different backgrounds does not allow to completely develop a complex software project. Are there aspects of HCI that are specifically related to this type of projects and would not emerge in the study of simple products?
3. I run the class with juniors and seniors (third and fourth year students) and I found that having "mature" students helped a lot with discussions, independent studies and the final project. I would like to know about the experiences that other teachers might have had in teaching younger students.