HCI education: We are failing – why?

Alistair D N Edwards, Peter Wright & Helen Petrie
Department of Computer Science
University of York
Heslington
York YO10 5DD UK
+44 1904 432775
Alistair.Edwards, Peter.Wright, Helen.Petrie@cs.york.ac.uk

ABSTRACT

HCI as it is taught is failing. The ‘wealth’ of poor interfaces in everyday use is testament to that fact. In this paper we suggest that this because of the way that we teach, the right messages are is not getting through to the students who matter – the ones who are going to go on to implement software. We present some suggestions as to how to achieve better results.

Keywords
Human-computer interaction, teaching, Computer Science, Psychology, education.

1. WE ARE FAILING

In any HCI course there are some basic principles and guidelines that we would hope to convey to students so that they will apply them in the design of any interfaces they develop. In an assessment we would mark down an interface design which made it easy for the user to enter an order for $x$ items at a cost of $y$ pounds each when the user’s intention was to order $y$ items at $x$ pounds. Further marks would be lost if once the user had made such an error, they were unable to undo it. So it must be that the designer of the trading system for the Tokyo Stock Exchange evidently did not attend a suitable HCI course, or if they did, they did not pay attention – because they designed the system whereby a trader sold 610,000 shares for 1 Yen (1 Yen is worth about 0.5 pence), instead of one share for 610,000 Yen – and was unable to cancel the transaction. [1]

Another example of the effects of a poorly designed interface is the one which has been blamed for (or credited with) winning the 2012 Olympics for London, when a representative accidentally voted for Paris instead of Madrid (BBC, 2005b), thereby eliminating Madrid from the final round of votes. These are spectacular and widely publicized examples but there are many more everyday instances – less expensive but often annoying. Witness:

1. The numbers of papers published in the HCI literature which are little more than criticisms of interface designs (e.g. [3]; [4]).
2. A best-selling book [5] consists of little more than a list of (elementary) errors found in commercially available interfaces – many of them in software manufactured by the world’s largest software company.

It was suggested above that the responsible designers either never attended or did not pay attention to HCI lessons, but that is the real point: One way or another HCI education is not working, is not teaching the lessons that are required to the people who need them.

The evidence from student feedback on HCI modules taught in the Computer Science Department at York, suggests there are two main attitudes to the material:

1. It is too easy. Compared to ‘formal’ and ‘hard’ subjects such as hardware or formal methods. ‘Hard’ seems to be used in two meanings: solid and difficult.
2. It is all common sense.

A counter argument would be that the continuing success of the British HCI Group, its HCI conference, the UPA and the American counterparts are a sign that HCI is in fact succeeding as a discipline. Yet, as suggested above, a steady theme in HCI and Chi conferences has been the failure of interface designs and how often do we get the feeling of preaching to the converted? Alternatively, as Bill Buxton recently put it, ‘While the CHI community is fiddling around with our ethnography-usability dilemma, Rome is burning, and has been doing for years!’ [6].

---

1 Note that, in the style of old-fashioned Japanese chivalry, the President of the Stock Exchange took the blame for the error and resigned – not the interface designer. It is also worthy of note, though, that this is not the first time such an error occurred in the same stock exchange. An almost identical mistake was made in 2001.
Another factor is that interface design is much harder to notice when it is done well. Apple has survived by designing credibility. Teaching tool is that it has gone too far. The resulting application is so unusable that it is impossible to use and lacks available) which has been designed by breaking as many usability heuristics as possible. The problem with it as a experience of poor interface design, but they may be unaware of it because they do not perceive of it as being poor. At the same time, subjects such as Maths and programming can be very hard intellectually. There is a satisfaction in studying a topic which is hard to understand but which becomes clear with time and effort. The effort can lead to an awareness of a level of mastery. A student may say ‘I know how to program in C’, but who would every say ‘I know how to design interfaces”? Again, though, with Maths and programming the student knows when the subject matter has become clear, when the answers they derive match the ones in the back of the book. With HCI there is not a lot one can learn by rote and there are no hard-and-fast rules that have to be applied in a given situation. This is what leads to the second reaction.

3. IT IS ALL COMMON SENSE
A problem we have is that HCI is inevitably multi-disciplinary. In particular, it involves an understanding of Psychology. Within a short course on HCI it is not possible to give more than a brief introduction to Psychology, so that the students tend to get a superficial impression – which they find hard to distinguish from ‘common sense’. For instance, we attempt to make it simple for students by distilling HCI wisdom into simple bite-sized chunks, such as Shneiderman’s Eight Golden Rules [7]. Take the first of these as an example: Strive for consistency. At first glance that seems indistinguishable from common sense and in a busy curriculum we do not have the time to encourage the students to look more broadly, to look at Grudin’s Case against user interface consistency [8]. If they did they would find that there is rather more to the question, that ‘consistency’ is not a simple, obvious quality – which explains why many interfaces are not as consistent as we might expect.

4. WHAT CAN WE DO?
The point that many interfaces are poor is not getting across. Students use a lot of software and so inevitably have experience of poor interface design, but they may be unaware of it because they do not perceive of it as being poor. There is a self-perpetuating cycle. Programmers design interfaces that suit themselves. When other (student) programmers use it, it matches their style of interaction. In other words, as [9] (p.3) has suggested, ‘the human interface to some software applications gives the impression that the designer’s model of the user as a 25-year-old male with a doctorate in computer science who is besotted with the technology and is more interested in playing with a computer than in completing a useful job of work!’ This willingness to play is another factor. Programmers are natural problem-solvers who enjoy the challenge of finding work-arounds more than they would enjoy a straightforward, intuitive interaction.

One attempt to address this blindness to poor quality has been attempted by IBM. EasyChart2 is an application (freely available) which has been designed by breaking as many usability heuristics as possible. The problem with it as a teaching tool is that it has gone too far. The resulting application is so unusable that it is impossible to use and lacks credibility.

Another factor is that interface design is much harder to notice when it is done well. Apple has survived by designing usability. In the early days of personal computers, the Macintosh never got anywhere near the PC in terms of sales. It was a lot more expensive than the PC – and yet it sold in sufficient numbers to keep the company alive. Now that company is thriving on the sales of the iPod. Owners and adverts extol its virtues in terms of the technology, the numbers of tunes that it can hold, the attractiveness of its design. Its usability is just taken for granted. Even the hardest Computer Science student, if asked, would appreciate the usability – but not the effort which went into achieving it.

Perhaps we are too soft on students and before we try to teach them any HCI we could give them a foundation of ‘real’ psychology? More emphasis on analysing and criticising existing systems might give them more insight and using other students’ designs might reinforce the lesson.

Also, teaching and assessment may have a tendency to focus on the outcome – and not the process. This can reinforce the perception that some of these students already have. To them the fact that a program works is more important than how it was developed. Thus, when confronted with a question which has no right-and-wrong answer, where a well-put argument in either direction will attract equally good marks, they are not equipped to cope. If we were to teach more about the design process and design rationale [10], they might be better equipped to understand the nature of interface design and hence more receptive to the material in an HCI course.

4. SUMMARY
The point has been made above that it is easier to criticise poor interface design. It is similarly easier to be negative about the way we teach; suggesting better ways is rather more difficult. Nevertheless, there is no doubt that there is vast scope for improvement in HCI teaching and this paper is offered as a starting point towards finding ways of achieving improvements.

References