Teaching and learning HCI

Harold Thimbleby
Future Interaction Technology Lab
Swansea University, Wales

Abstract. We consider how to teach effectively with particular reference to HCI. HCI can be taught to explicitly empower students to engage with their own learning. Further, HCI motivates because HCI empowers students to make a valuable and lasting contribution to the world.

Keywords: Teaching and learning. HCI (human-computer interaction).

"The main part of intellectual education is not the acquisition of facts but learning how to make facts live."
Oliver Wendell Holmes, Jr.

Cite as: H. Thimbleby, "Teaching and Learning HCI," *Proceedings HCI International, Part I, Universal Access, HCII 2009*, Lecture Notes in Computer Science, **5614**, pp.625–635, edited by C. Stephanidis, San Diego, Springer Verlag, doi:10.1007/978-3-642-02707-9_71, 2009.

1 Introduction

The world could be a better place, and of all the things that need improving, user interfaces should be near the top of the list, because bad design of user interfaces makes many other things and experiences worse. A bad user interface for a web site can cause users to make expensive errors; or a bad car radio design can distract a driver from attending to the road that they have an accident. A badly designed user interface for a hospital drug delivery system (an infusion pump) can induce nurses to make fatal drug overdoses. A badly designed web site can detrimentally influence millions of people: it has a huge and hidden social cost. Indeed many user interfaces are bad, and their faults are so obvious — at least to those who know HCI and have the background, inclination and education to *see* the faults — they clearly ought to be taken as a point of high leverage to invest in to improve quality of life. **The most effective way to improve HCI is by improving HCI education**, for each educated and motivated HCI student will go on to have many opportunities to continue to improve user interfaces far into the future, and improved user interfaces will leverage improved experiences for all their users. (If you do not believe user interfaces are bad, please read *Press On* [12].)

Many user interfaces are bad yet we have the processes and knowledge to do better, therefore HCI *education* must have failed the developers or marketing people or managers who create and distribute the current poor systems — it has certainly failed the users of these systems, the people affected detrimentally by them. At face value, then, the problems of HCI are symptoms of a lack of appropriate and effective

education. The people who know HCI are somehow not in the right place to use it, or the people who need HCI are unaware of it, or the HCI that has been taught has not motivated students sufficiently to apply their knowledge. Perhaps we have only taught students to pass exams and not to use their knowledge in the world of work?

Unfortunately, questions about education and understanding a subject are rarely addressed in the academic or research literature about that subject.

Consider the instructions for writing ACM papers, the ACM being the leading international computing organization. The ACM computing classification system gives an explicit list of topics, but these classifications do not expect articles that talk about how any subject is acquired, understood, used or taught; they expect topics like "human factors," not the topics of *thinking about or reflecting about* "human factors," whether teaching, communicating, or even using it. It is as if just stating facts are sufficient, as if nobody needs to think about how facts are presented or learnt, whether by researchers or by students, or even how such facts may be effectively communicated from author to readers, and how those ideas are then applied to change the world. Ironically, while computer science includes topics like communication (and HCI human-human and human-computer communication, *etc*) we ignore communication specifically to communicate human knowledge about our own areas of work! (This view will be encountered again, below, as an expression of Ramsden's Theory 1; and as the concept of interactional expertise.)

Kline [7] presents many ways that our academic culture undervalues pedagogy — pedagogy being one way of understanding and thinking about a subject. For example, while many organizations and industries have research arms, many universities have no research in-house into one of their core activities, specifically teaching. (Many universities research education, but they do so academically, not to enhance the university's own teaching processes.)

How then should we teach and think about teaching HCI? Teaching is the highest form of understanding; if we do not understand how to teach, we do not understand our subject. If we are not thinking about teaching, we are not thinking about communicating. Even the most hardened researchers must surely be concerned about the impact their research papers have; in fact, their research papers must surely aim to teach their readers new ideas and new ways of thinking about their subject. This isn't so different from wanting to teach students. Ironically, many researchers see prestige in reaching *fewer* researchers, rather than in being accessible to *more* students.

According to Ramsden's excellent survey [10], teachers (for instance, teachers of HCI) consider there are three approaches: **Theory 1** teaching is telling or transmitting facts; **Theory 2** teaching is organizing student activity; and, **Theory 3** teaching is about making learning possible.

That is, teachers adopt a tacit stance to teaching and learning, which can be put into one of these three classes depending on their approach. Moreover, students adopt a complementary approach, and subsequently the teacher's tacit views are reinforced as they teach better to meet the expectations they have created.

Many HCI textbooks are encyclopedias of knowledge about HCI techniques, as if their authors fall into a Theory 1 approach, into a style that supports an assumed Theory 1 style of teaching. The teacher's job, using such books, is to teach the students the facts of HCI, preferably as presented in the particular books chosen. The students know they will be examined on these facts, and they demand clearer teaching

of those facts. Theory 1 encourages a style of thinking that every fact must be covered, and that it is the teacher in conjunction with the textbook author whose job it is to provide all facts that need teaching.

Different subjects and different stages of learning in those subjects call for different approaches. In an early anatomy or geography course there are indeed a lot of independent facts to learn, but these facts give way to deeper learning as the student progresses. Similarly in HCI, there are indeed many important facts to learn — what is affordance? what is contextual design? what is immersion? what are ethical experiments? — before one can build deeper knowledge and understanding.

There are of course many areas and stages of learning where Ramsden's Theory 1 may be entirely appropriate, most obviously in areas with many basic facts (e.g., anatomy) and at elementary levels when the student is not expected to need real understanding of the subject but needs to learn the definitions. This a student might be taught to "always end a sentence with a full stop." At school, there may be no room for debate on this fact. Yet when the student becomes a designer, they will discover that posters often have sentences without full stops, and that one can decide, not on rigid grammar, but on (for example) visual criteria, or on unrelated non-grammatical criteria such as whether your client will pay. Because language is necessarily first taught in elementary ways to young learners, many of us have grown up thinking that our use of language is rigidly constrained by what we were taught [3]. We've learnt (mostly in childhood) that it's *just* non-negotiable rules — unrelated facts. Perhaps this formative learning experience, learning language, has influenced our approach to other learning and teaching?

Theory 1 is necessary, but it is not always sufficient. We spent many of our formative years being taught elementary facts, and it is understandable how we end up ourselves being teachers who emphasize facts. If we are not careful, we end up with students who know some facts, namely, exactly the ones we teach them to pass their courses — but they don't know how to think for themselves about HCI, and are therefore unable to apply their knowledge to the work environment they later find themselves in. Ultimately, as students graduate and get jobs, we end up with interactive systems — web sites, ticket machines, voice menus, aircraft, medical equipment — that have bad user interfaces. Or as students graduate and become academics, their views influence how they participate in the academic community: they become referees (of research) and teachers (of students). The Theory 1 attitude affects referees for research papers and research proposals [11]: a common criticism in HCI refereeing is that some *facts* or references (pointers to facts) were missing (i.e., facts from a different subdomain of HCI that the referee wishes to emphasize), as opposed to some *reasoning* was flawed.

Collins and Evans [2] make a related distinction: there are two sorts of expertise as outcomes of teaching: *contributory expertise* and *interactional expertise*. Interactional expertise is the knowledge and ability to talk about a subject, perhaps passing off as real expertise, whereas contributory expertise also has the skills and know-how to work in or contribute to the field. Clearly, we want students with growing contributory expertise, rather than just the interactional expertise sufficient to pass assessments. In these terms, our failure is to teach students who pass off interactional expertise (which we assess) as contributory expertise (which they need to work, when they go on to design and evaluate user interfaces).

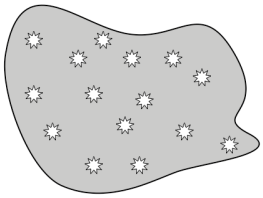

Fig 1. A visualization of all known HCI concepts, represented as stars.

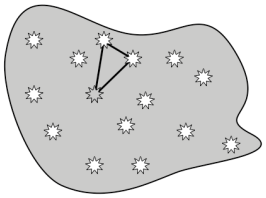

Fig 2. Forgotten facts (represented by a few missing stars) may be triangulated from known facts.

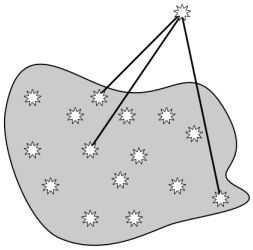

Fig 3. New discoveries extending existing knowledge are made by triangulating from the known to the unknown.

Ultimately, then, Theory 1 is not an effective form of teaching for HCI, at least if we want students to contribute to improving the world. Indeed, Ramsden makes it clear that Theory 3 is, for most things, better.

An example of Theory 3 teaching comes from Feynman [5], who is widely recognized as one of the most inspiring teachers of physics. Figure 1 shows an imaginary map of all HCI ideas and concepts within a region drawn as a grey blob; figure 2 shows how a student might be able to reconstruct a forgotten fact from several other remembered facts. Probably a student would use some remembered facts, some books, and some experiments: triangulation is somewhat of a simplification to the idea. (In reality, HCI is so complex that lots of facts would be needed to triangulate, and perhaps the idea might better be called interpolation.)

The purpose of teaching a student is so that they are eventually able to construct new knowledge — it would be a sorry state of affairs if they could only ever know less than their teacher! Figure 3 shows how the same triangulation idea works for a student discovering new knowledge. The point is, by teaching a student how to connect ideas together, they are empowered to learn new things, and even ideas they were not directly taught.

Feynman sees the blobs in these figures as knowledge, as potentially known by everyone. Instead the blobs might be used to represent the student's own knowledge. Then, that star in figure 3 might be some x a student missed. It is useful for a student to realize that x is missing and be able to work it out; this is far better than only knowing *at most* what they were taught from figure 1.

Perry's study of how students learn suggests that the least sophisticated students, students at early stages of learning (a point some students never progress beyond), want to learn *true* facts [9; 12]. Students at this level thus dovetail their expectations with a teacher's use of the Theory 1 approach to teaching. Unfortunately, both Theory 1 and Perry's low end of sophistication interact in a vicious circle: they support each other, and are ideal for teachers and students with little confidence in the subject. Neither enables the students to go beyond the teacher, so the students are limited to exactly what is taught. Students soon will only do work that leads to assessment. It is but a short step to automating the assessment, typically with multiple-choice questions, to see exactly what facts that the student has learned: once automated, the student is even denied any flexibility in interpreting the right answers.

Dweck identifies a personality dimension, performance—learning [4]. A performance- or outcome-oriented student wants to do well, and tends to avoid problems they find hard.. In contrast, a learning-oriented student prefers acquiring skills and understanding, and thus accepts making mistakes and under-performing in the short term. The dimension is interesting because it correlates with many other personality factors, attitudes that reinforce each other. Thus performance orientation has also been called "fixed mindset" versus the "growth mindset" of the learning orientation. A performance-oriented student believes they have a fixed natural ability (or not) to achieve in particular areas (like sport, mathematics, HCI); they are endowed with specific abilities such as intelligence, and thus they are either born good at a subject or born bad at it. They believe they will never be much good at things they find "naturally" hard. In contrast, a learning-oriented student revels in challenges, and believes that abilities, such as intelligence, can be developed.

In formal education, students are generally assessed on performance rather than learning. It suits educators in many ways; performance is easier to assess than learning, and if students do badly, a teacher can hide behind a performance model: the students were bad anyway. In contrast, if a learning-oriented teacher has poor students, this is a challenge to try to help them find out how to learn better.

Dweck has shown that rewarding performance can make students worse when later work is more challenging and requires more effort; only praising good performance weakens a student's resolve to persevere through later failure. Learned helplessness is therefore a real problem for performance-oriented students: rather than risk not looking smart and risk the failure of poor performance (e.g., on an assessment) they sabotage themselves so their poor performance is due to some non-intellectual or less-personal factor, such as disorganization. If they put little effort in to a task, either they will do well (because they are naturally good), or they will do poorly (because they didn't try): it is thus safer for a performance-oriented student to learn helplessness in the face of anticipated weak performance or required effort.

Dweck's psychological model supports the descriptive views of Ramsden's Theory 3 and Feynman's triangulation. Performance-oriented students are threatened by the success of other students and they can therefore set out to drag a class down to their level. On the other hand, learning-oriented students are inspired by other people's success. It is important for a teacher to nurture learning and an *attitude* to embrace learning. Fortunately few real problems in HCI have known solutions — both teacher and students have to find out the answers by doing experiments. This puts the teacher on a level with students, develops learning-orientation, and encourages peer learning.

2 Teaching HCI *as a form of HCI*

Teaching is about getting students to learn and engage with ideas, ultimately to own them. Correspondingly, we can consider that HCI is concerned with getting users to learn and engage with ideas about interactive systems. It's the same thing. Research in HCI, such as Carroll's classic work on "minimalism" [1], suggests that users are best helped when instruction or training follows four principles:

1 Choose an action-oriented approach; provide immediate opportunity to act;
2 Anchor the ideas in a task domain; select real tasks;

3 Support error recognition and recovery; prevent mistakes where possible;
4 Support reading to do, study and locate; be brief — don't spell out everything.

Of course Carroll elaborates these principles further, but even from this brief summary, it is interesting to note that Carroll's principle 4 manages to simultaneously contradict the "bad" Theory 1 teaching and support the "good" Theory 3 teaching. HCI itself suggests that HCI *teaching* should provide an immediate opportunity to act, based on real tasks, should prevent mistakes, and be brief. Indeed, a significant part of any HCI syllabus should be user learning, a topic that makes a useful counterpoint to students learning HCI, and is an opportunity for students to be taught learning and learning skills explicitly, something that, sadly, few have encountered in their higher education. Carroll is only one example; as HCI is concerned with the user experience, almost any HCI issue begs an analogy to teaching and learning. Dweck's notions of performance and learning orientation relate not just to students but also to users who have to decide whether to achieve results quickly or whether to learn more advanced features of a system so they can do even better, but by delaying immediate results.

Kline suggests that the worst sort of teaching presents unmotivated facts [7] (cf Ramsden's Theory 1); the facts may be motivated for the teacher, but to the student they seem pointless. Carroll's point 3, above, that suggests that students might learn by making mistakes and learning from them. It is fascinating to draw parallels between students' learning and mistakes with users' learning and mistakes: there are plenty of stories of aircraft accidents, clinical incidents ([6] is a highly-motivating resource, including Human Factors studies), and other disasters that illustrate HCI issues. Even my own fumbling with projectors and computers can be recruited to illustrate HCI issues, and certainly when students do presentations to the class their inevitable problems with equipment can very usefully be turned around to explore the latent errors in the design of the equipment they are trying to use under the real pressure of presenting to their peers.

Theory 1 does not do well from Carroll's perspective either. It is hard for a student to make a mistake when their teacher adopts Theory 1 and for them not to be simply wrong; there is no incentive to learn from mistakes. Worse, as Theory 1 approach leads to simple assessment, it's likely that the only feedback students get on their understanding is when they are formally assessed: a significant disincentive to make mistakes or even explore around the subject. Why would a learner experiment, possibly making mistakes, when doing so guarantees getting fewer marks?

Ong suggests [8] that ever since the invention of the alphabet (one of the earliest technologies) we have taken it for granted that knowledge can be written down. Further, if it can be written down, we can teach what is written. But that is Theory 1. Notice how it seems obvious we must do this, for how else (it seems) can the knowledge of writing be preserved?

Rather, ask why do we lecture when we have writing, and books in particular? The answer is that we should not teach facts, for that encourages shallow learning. Instead, we need to motivate, make accessible, *enthuse*. As Carroll's work suggests, we need to get students engaged with real tasks as quickly as possible. Why do students go to lectures when they could read books or read off the web? Somehow the interaction and excitement of the lecturer is supposed to rub off in a way that the textualized book or web page does not permit. Teaching is performance (in the theatrical sense), not just instilling facts.

Most universities have adopted computerized teaching systems. These provide ideal HCI hunting grounds, and in exploring them students not only learn to think about HCI in a domain that affects them, but they also learn more about learning.

3 Personal values

We, whether students or teachers, are all different and we all have different perspectives to bring to the teaching and learning forum. As teachers, we have had formative experiences as students ourselves, and sometimes we tacitly emphasize personal values rather than ones supported by good pedagogy. I am no exception.

Here are some values I consider very important. My experience, as is obvious influences the experience of my students and even the students who choose to come on my courses. This biased sampling reinforces my prejudices! You do not need to agree with me, but I think you need to work out your own framework.

Teaching and learning is fun. If people are not enjoying what they are doing, this in itself is demotivating. If students enjoy their work, they will do it better, they will be more committed to working on it, working hard, and thinking deeply about it. And work that a student has done that they have enjoyed will be more enjoyable to mark.

Teaching and learning is fire. It's not just fun, it's serious fun: fire in our hearts, fire that spreads, fire that lights the imagination. It's about things we feel strongly about — nothing luke warm. By teaching we light up students and are more effective than in ordinary jobs where we would have no such leverage. In each class we want to inspire especially those students who are sparked by the subject and are going to carry the flames forward.

Teaching and learning is exploration. I know the terrain, but I want students to find things out for themselves, and especially find out things I don't know. Because my lectures are interactive, students like leading me down garden paths. The students think they are distracting me; but I know we are exploring the HCI issues of what they are interested in. In HCI we are lucky, for there are so many unknowns, and so many new questions that can be answered by student exploration.

Teaching and learning are interactive. I do not go into lectures to tell students what I know; I go in to enthuse them, and that means finding our what they want to know, and apprenticing them with me to have deeper understanding, as I take what they know into areas they do not yet know they need to know. I am very reluctant to provide handouts, because the handouts are not the lectures; I have very few facts on slides used in lectures — mostly they are pictures that create discussion, and help organize my lecture so I cover planned topics. But a student who fails to come to a lecture and wants "the notes" won't be able to tell whether the slides are examples, say, of good or bad practice!

There is too much in HCI to teach in any single course. I do not presume that what I like in HCI is what will engage my students, and the early parts of my courses involves negotiating with the students what they will learn and engage with. I hope they want to learn about my preferred topics, of course, but if they want to learn about (say) CSCW, I'd rather help them learn that and get involved with the subject than just go through the motions of teaching them my agenda but they never properly engage again with it in the rest of their lives. Because my courses not assessed by

exams, I do not need to teach to a prior syllabus; and if I have to use exams, then I want to set the exams *after* this period of negotiation.

Because I take this approach, my lectures potentially do not teach enough facts. I repeatedly emphasize that students have to take responsibility for learning facts: there are plenty of good books. Part of my teaching therefore covers reviewing the strengths and weaknesses of relevant books, including my own [12].

Teaching and learning is research. The students can find out things, test ideas, and find out things none of us knew to start with. The problem with this style of teaching is that it is hard to predetermine outcomes; it is certainly nearly impossible to provide notes beforehand. (But notes fall into Ramsden's Theory 1 view.) This stance makes it particularly difficult to support students with special needs who benefit from polished material and material presented in different media. Nevertheless, any special needs are a problem for the *whole* class (and any helpers), not just the teacher, and — most especially in HCI — initial homework for students can be to search for solutions and work out how to implement them in the session.

Teaching and learning encourages mistakes. Educational environments are a safe place to learn about recognizing and managing errors: if you are not dropping the balls, in juggling, you aren't learning anything. This short paper has said little about syllabus/curriculum, but a very important area that needs emphasizing (especially in HCI) is testing, user testing and debugging — iterative design and design discovery. Most HCI problems would have been avoided if their designers/developers had learned that systems are *always* inadequate and need further testing.

Teaching and learning is formative. The students want feedback about their achievements; I want feedback from them about my teaching — and both of us want it formatively, not at the end of the course. I believe I can do better, and I encourage students to give me feedback, to point out mistakes or things they like. Indeed, as Carroll [1] wanted with user training, if students can recognize my mistakes, I am lifting them from passive learning to active participation. By discussing mistakes in lectures, they are learning much more useful attitudes and skills.

Teaching and learning is open. There are many horrible arguments, real and imagined, for being secretive about teaching and learning. Failure is private. Success causes envy. People may steal my good ideas. It surprises me how rarely teachers share insights into each other's work — even if they know about it. I feel I am *intruding* when I go to other lecturer's classes! (And no colleague has been to one of my own classes for a long time.)

I am increasingly assessing students in open ways: for example, asking them to do coursework as posters, not essays. Then an afternoon's poster conference can both have me marking the coursework (and interacting helpfully with the students at the same time), as the system requires, but more importantly each student sees the quality of each other student's work. They learn by my creating open processes.

Teaching and learning is reflective. In addition to the obvious HCI content, I also teach how I teach and why I choose particular approaches, and I teach how students may learn better, and I do this within the course. We all then engage consciously with the teaching and learning process, and renegotiate changes each time I teach. I encourage students to think explicitly about how they want to be successful. I ask them to review and analyze which courses and styles they find helpful.

This approach to teaching is analogous to putting iterative design into practice, except I apply the benefits of formative evaluation for teaching. The analogy works powerfully when combined with actually teaching iterative design or evaluation, for instance as part of a lecture on ISO13047.

Teaching and learning are paradoxical. Over my life, I have learnt many complex things, like speaking and walking, without anybody really trying to teach me; and I'm glad I learnt these things before school. School "taught" me lots of things I have not, in the end, learnt, and it put me off many other things, like dance. Conversely, I have taught many complex things by not trying to teach at all. My children know how to solder, but I didn't teach them in any way a university would recognize, with notes, assessments or planned learning outcomes; it was a lot easier than that, and they never said they'd only do it if I assessed them!

Teaching and learning don't stop. I want my students to learn more than I know, and certainly more than I can teach. One consequence of this view is that, where possible, I use coursework rather than examinations. With exams, there is a fixed syllabus represented by the questions, and at some stage you have to start playing games with the students: in a revision class, for instance, you can't *really* tell them the answers to the questions you've set. You get into complex political games, made worse by "marking schemes" and other processes for fair marking. With coursework (portfolios and other techniques) you as a teacher always want the students to do as well as possible, and there is no need to hold back on telling the answers — you want the students to know, so they can go beyond them. Conversely, the students don't ask, "do we need to learn this for the exams?" as anything and everything you teach can help in their coursework (and, later, in their real world work) — there are virtually no exams in the real world, so why train students to perform to them?

Less is more. I could extend this list indefinitely, but less is more. You, the reader, must surely have started to have your own (and better) ideas about teaching and learning, and if I carry on with my ideas you will lose your own ideas. Similarly, if we put more effort in to teaching, say, writing detailed notes, the less we leave for the students, and the less space we leave for our lecturing to unfold in the dynamic relationship we create with the students. Then, the less the students own of what we teach, and the less they learn the gift to do things we didn't teach them explicitly.

4 Conclusions

A proper concern of *any* subject is how people learn that subject, for if they do not learn it successfully, then the subject fails — certainly the academic community fails. If the subject is too obfuscated, uninteresting, dead, then it becomes at best the isolated thinking of the few. HCI subject needs to be successful in the world: it needs practitioners who understand and apply and contribute to the subject. We therefore have to focus on pedagogy as a proper part of the discipline. To make the world better we want our students to engage and become contributory experts [2] who *actually* contribute; this coincides with Kline's, Feynman's, Ramsden's and Dweck's views.

HCI is concerned with how people learn to use complex systems effectively. A lot of HCI is fun, but a lot of it is crucial, both for manufacturers to stay competitive, and for users to stay safe. Many issues in HCI can also be presented as reflections on how

HCI itself is taught; HCI is a complex system, and students are users of it. Am I teaching HCI in a way that is compatible with what I am teaching about good HCI practice? There is debate to be had, and students can get into it and start thinking, doing experiments, and triangulating new ideas from what they are learning. It is easy to show students that they are starting to learn important, life and death things that the world needs to know and to apply.

How do we teach HCI? My answer is to enthuse students with the enormous impact HCI can make to the quality of life around them *and* to teach them about learning. HCI itself is well-suited to this "metateaching," as one of its core concerns is user learning. HCI is a subject that is everywhere, even in the classroom. Even when the projector doesn't work, perhaps *especially* when the teacher despairs with the projector's terrible HCI, then HCI becomes relevant and alive to the students.

HCI is a subject with a crucial role in quality of life so we should take it seriously. It amazes me that taking things seriously, particularly in higher education, often leads to us making things private and unexciting. On the contrary, HCI begs to be public and exciting. Why do we hide academic results (and get bored) but get excited over football games, where success and failure are public? People strive to get better when they get excited, and frankly most students fail to work out how to get excited over anything that is as private and secretive as conventional education has become. As teachers we have a pleasurable duty to work out with our students what is exciting. Making HCI public is the best way to improve the world.

Acknowledgements. Harold Thimbleby is a Royal Society-Leverhulme Trust Senior Research Fellow, and gratefully acknowledges this generous support. This paper is a revised version of [13], "Teaching HCI to make it come alive," by permission of the author. Greg Abowd (Geogia Tech), Alan Blackwell (Cambridge University), Paul Cairns (University of York), and Tony Hoare (Microsoft) made extremely valuable comments, for which I am very grateful.

References

1. Carroll, J.M. ed.: *Minimalism: Beyond the Nurnberg Funnel*, MIT Press (1998)
2. Collins, H. & Evans, R.: *Rethinking expertise*, University of Chicago Press (2007)
3. Crystal, D.: *The fight for English*, Oxford University Press (2006)
4. Dweck, C.S.: *Self-theories: Their role in motivation, personality, and development*, Psychology Press (2000)
5. Feynman, R.P., Gottlieb, M.A. & Leighton, R.: *Feynman's tips on physics*, Addison-Wesley (2006)
6. ISMP, Institute for Safe Medication Practices, Canada: *Fluorouracil Incident Root Cause Analysis*. http://www.ismp-canada.org (2007)
7. Kline, M.: *Why the professor can't teach*, St Martin's Press (1977)
8. Ong, W.: *Orality and literacy: The technologizing of the word*, Methuen (1982)
9. Perry, W.G.: *Forms of ethical and intellectual development in the college years*, Jossey-Bass (1999)
10. Ramsden, P.: *Learning to teach in higher education*, 2nd ed., RoutledgeFarmer (2003)
11. Thimbleby, H.: "Supporting diverse HCI research," *Proceedings BCS HCI Conference*, **2**, pp125-128, edited by Dearden, A. & Watts, L., Research Press International (2000)
12. Thimbleby, H.: *Press on: Principles of interaction programming*, MIT Press (2007)
13. Thimbleby, H.: "Teaching HCI to make it come alive," HCIed2008 Conference, *Interaction Design and Architecture*, **3–4**:9–16, special issue edited by Giovannella, C., Kotze, P. & Wong, W. (2008)