

2-17-2011

Preparing Computer Science Graduates for the 21st Century

Paul Parsons

The University of Western Ontario, pparsons@uwo.ca

Follow this and additional works at: <http://ir.lib.uwo.ca/tips>

 Part of the [Computer Sciences Commons](#), and the [Higher Education and Teaching Commons](#)

Recommended Citation

Parsons, Paul (2011) "Preparing Computer Science Graduates for the 21st Century," *Teaching Innovation Projects*: Vol. 1: Iss. 1, Article 8.

Available at: <http://ir.lib.uwo.ca/tips/vol1/iss1/8>

This Article is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Teaching Innovation Projects by an authorized administrator of Scholarship@Western. For more information, please contact [Natasha Patrito Hannon](#).

Preparing Computer Science Graduates for the 21st Century

Summary

The nature of computer use has changed remarkably in the past fifty years. However, most undergraduate computer science courses are still often taught through an old paradigm that is not adequate to address modern concerns. This 90 minute seminar will address some issues relevant to preparing computer scientists for the 21st century. These include issues central to human-computer interaction (HCI) such as cognitive and perceptual aspects of computer users, ergonomics, and human factors. Although there has been literature on this topic for at least the past 15 years, it is still not widely recognized nor understood by the majority of computer science educators. Computer science graduates are often expected to have an understanding of many issues surrounding the interaction between humans and computers when they are in the workplace. However, most computer science graduates are ill equipped to deal with such issues, and could benefit if they were given more consideration in the university curriculum. In recent years, interest in HCI has grown enormously in both industry and academia. The Association for Computing Machinery (ACM) recently reported that its special interest group in HCI is the fastest growing of all its interest groups, and has recommended the development of new HCI programs in universities to combat a shortage of professionals with the skills and training to advance the design of more usable technologies.

Talking about this issue can hopefully arouse awareness among computer science educators about its importance. Additionally it is hoped that seminar participants will be able to understand some of the main issues surrounding HCI teaching and education and how to begin to address them. The seminar will examine a number of contemporary issues regarding computer science education and what experts are saying about it.

Keywords

learning, engagement, computer science

Creative Commons License



This work is licensed under a [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/).

Objectives

By the end of this session participants should be able to:

- Identify HCI concepts and their relation to computer science education
- Relate cognitive psychology principles to the use of computer systems
- Evaluate their teaching curriculum and its strengths and weaknesses in relation to human-centered issues
- Identify some strategies for teaching HCI issues at an undergraduate level

Reference Summaries

Cockburn & Bell – Extending HCI in the computer science curriculum

In this paper Cockburn and Bell describe some of the ways that they have been integrating HCI into their computer science curriculum at the University of Canterbury. They describe their motivation, some of the techniques they have used, and some of the benefits that they have found or expect to find. The authors suggest that computer science students should be taught topics such as elementary psychology, ergonomics, experiment design, and evaluation techniques. They argue that this will lead to better prepared graduates who are capable of handling current and relevant issues. This paper is useful for the seminar since it provides a motivation and rationale from experts in the field. This will help to validate the importance of the seminar topics to the participants.

Greenberg - Teaching human computer interaction to programmers

In this paper Greenberg explains a course he developed for computer science undergraduates. The course introduces students to some of the human issues surrounding the design of computer software. The rationale is that computer science graduates are often expected to be able to design software interfaces but are ill-prepared with a traditional computer science education. This paper will be quite useful for the seminar since it is about an educator's experience trying to integrate HCI concepts into traditional computer science education. I can use his examples as well as the things he found to work well as a guide for the participants in the seminar.

Douglas et al. - Incorporating Human-Computer Interaction into the Undergraduate Computer Science Curriculum

This paper presents issues and experiences related to incorporating HCI issues into the computer science curriculum. The authors state that the majority of computer scientists have been slow to adopt such issues into the curriculum even though they are of critical importance. Although many computer science educators have begun to recognize the importance of these issues, only 40% of programs offer a course. One of the authors suggests making HCI a required course for undergraduates. Another author suggests that this type of training will prepare computer scientists to communicate and work with designers. This paper is useful since it provides a motivation for the topic. It also provides some statistics which highlight the need for educational change. I will use some of these statistics in the seminar to show the participants why this is an important issue.

Hewett et al. – ACM SIGCHI curricula for human-computer interaction

This paper presents a report assembled by the SIGCHI (Special Interest Group in Computer-Human Interaction) Executive Committee on curriculum recommendations for education in computer science. The rationale for the compilation of this report is that there is a general shortage of educational materials for preparing undergraduate courses dealing with HCI issues. The report provides an account of the historical roots of HCI and its relation to other areas such as computer graphics, cognitive psychology, human factors, and industrial engineering. The report also briefly discusses each of these supporting fields. The course also proposes a number of different possible undergraduate course designs, each with slightly different focuses. This paper is much more detailed than the others. Unlike the others it provides a number of different educational approaches, each of which has different benefits. I would like to use these in the third portion of the seminar to generate some discussion and get some feedback from participants.

Scown & McManus – Educating computer people about people & computers

This report describes a panel session where a number of experts discussed the key issues affecting the teaching of HCI. Some of the issues discussed were: how can educators cope with the flood of new HCI ideas? Should HCI be fully integrated into software engineering courses or taught in specialist modules? What are the core elements of HCI that all students should learn? There are perceived cultural differences in HCI teaching in the UK, USA and Scandinavia -- are these differences real and are they important? Should we be sharing educational materials? Can HCI design be taught or are creative designers born not bred? This paper was useful for getting an idea of some of the main issues that educators are facing, and how they deal with them. I will use some of the responses in the paper to guide the third part of the seminar (see below).

Content and Organization

The following table provides a breakdown of the 90 minute seminar. The first column contains the approximate timing; the second column contains the content for that section of time; and the third column contains examples and other materials used to support the content.

0-15	Introduction – a general overview of the topic; how has computer use changed over the past 50 years; has computer science education kept up with the changes in the way computers are used; why should we as educators care about human issues in the context of computer science; what are some of the consequences of not taking these issues into consideration;	In this section I will use a number of general but real-world examples to back up my ideas. For instance, when discussing why these ideas are important I will provide some examples of disasters that have occurred because human factors were not taken into consideration – e.g., three mile island disaster, Therac 25 accident, John Denver’s plane crash. All of these have been linked to poor interface design and a lack of understanding of human issues on the part of the designers. These examples will help the seminar participants to see that these are real and serious issues.
15-45	Group discussion – any general comments about the issues we just talked about; have people heard of these before; do we see	In this section I won’t be using as many examples as the previous section. I would want the participants to discuss the

	why these are important; offer any suggestions as to how you have addressed any of these issues in your teaching; do you think we should be changing computer science education to address some of these issues or is it fine the way it is; if you have addressed these in your own teaching, what worked and what didn't work; what other suggestions can you offer to computer science educators from your own experience;	examples mentioned above.
45-75	Case Studies & Role Playing with a focus real-world teaching examples. Break the participants into 4-5 groups each with their own case study involving a scenario. Ask participants to come up with specific teaching strategies that might be useful and how it could be used. Each group will then display the case using role playing.	This section will be primarily using examples from educators who have attempted to integrate some of these issues into their computer science curriculum. The references above offer many different examples. During this portion each group builds a suitable teaching plan for their case study and then role plays it for the rest of the participants
75-90	Conclusion – what were the main issues we talked about in this session; what were the learning objectives for this session; any questions or general comments about the seminar;	This section will not introduce any new examples, but will briefly mention some of the ones listed above.

Presentation Strategies

In this seminar I will use a combination of presentation strategies. For the first part of the session, I would take a transmission approach, speaking about the topic in general and its importance. This will be primarily a lecture-style approach; however I would ask a few questions at the start to get some feedback from the audience. This will allow me to give participants some general information and to get them familiar with the topic. This portion would last approximately 15 minutes. I would also encourage participants to ask questions at any time, even during the lecture-style portions.

Once the participants become familiar with the general ideas I would then have a group discussion. Depending on the size of the group, I would like to hear from each person about what they know about the topic, what they perceive its importance to be, how they are teaching any of the concepts, and any other general ideas they have. I think it would also be beneficial to open the discussion up for comments and suggestions from all participants and not just from me. Doing so would allow participants to share their ideas in a way that they may have not had an opportunity to do before. As a consequence there may be novel ideas and teaching strategies that surface which can be beneficial to the whole group. This portion would last for approximately 30 minutes. This approach is in line with the developmental perspective on teaching that we learned about in ATP.

The next portion will go for about 30 minutes. During this time I would provide some case studies and guide the participants through them. I would address some of the aspects of teaching that are mentioned in the sources above and try and get the participants to apply theories to real-life scenarios. As such it will be a mixture of lecture and group discussion, but with a heavy emphasis on participation from the whole group.

The next portion will be a conclusion and will last for the final 15 minutes. During this time I would be doing most of the talking. The purpose of this portion would be to sum up the main ideas from the seminar so that when participants leave they have a clear understanding of what the seminar was about. I would address the learning objectives (mentioned above) and ask a few questions about them to make sure they are generally achieved. Towards the end I would open it up to questions and general comments. This type of conclusion will wrap everything up nicely for the participants so they walk away with a clear understanding of the seminar topics.

References

- Cockburn, A. & Bell, T. (1998). Extending HCI in the computer science curriculum. *Proceedings of the third Australasian conference on Computer science education - ACSE '98*.
- Douglas, S., Tremaine, M., Leventhal, L., Willis, C., & Manaris, B. (2002). Incorporating Human-Computer Interaction into the undergraduate computer science curriculum. *Proceedings of the 33rd SIGCSE technical symposium on Computer science education - SIGCSE '02*
- Greenberg, S. (1996). Teaching human computer interaction to programmers. *ACM SIGCHI Bulletin*, 28(2).
- Hewett, T.T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., & Verplank, W. (1996). *ACM SIGCHI curriculum for human-computer interaction*, ACM.
- Scown, P. & McManus, B. (1995). Educating computer people about people & computers. *Report on conference panel, HCI '95*.