HCI Graduate Education in a Traditional Computer Science Department

Saul Greenberg
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
+1 403 220 6087
saul@cpsc.ucalgary.ca

ABSTRACT
There are now several serious graduate programs dedicated to the training of professionals in Human Computer Interaction (HCI), and this is attracting considerable attention in the community. Yet HCI professors at most institutions are still limited to do this training within the constraints of a traditional department and program. In this paper, I discuss the issues that I and others encountered while creating an HCI program within a traditional computer science department, and my solutions to them.

Author Keywords
HCI Education. Graduate programs.

INTRODUCTION
Over the last few decades, educators in Human Computer Interaction (HCI) have legitimized this field as a necessary component of the computer science discipline in both undergraduate and graduate programs.

At the undergraduate level, introductory HCI material is now offered at many institutions. These typically appear either as fully-fledged HCI courses at the junior or senior level, or as components in other courses, e.g., within a software engineering course. There are several versions of HCI curricula, a variety of introductory text books, and good introductory lectures available on the World Wide Web. Some institutions even have a second specialized undergraduate course in HCI, or an 'HCI concentration' that suggests a slate of related courses to those interested in the area.

At the graduate level, the picture is somewhat mixed. At one extreme, there are still many institutions that have no faculty who specialize in HCI. This is not to say that these institutions intentionally neglect HCI, for the large number of faculty advertisement asking for HCI experience suggests that the bottleneck is acquiring HCI academics. At the other extreme are the few institutions that have created a formal HCI program. These programs solicit students who wish to become HCI professionals, and tend to encourage cross-discipline research and training. Between these two lies the more common situation where an HCI faculty or two craft a program or HCI concentration within the constraints of a traditional graduate computer science degree. By traditional, I mean that the degree program is primarily oriented toward the general discipline (e.g., computer science, psychology) vs. cross disciplinary, has breadth and depth course requirements specific to that discipline, and is thesis-oriented. Specialties within these degrees are usually by research interest of faculty and students rather than through formal program designation.

Greenberg, Saul. (2005) HCI Graduate Education in a Traditional Computer Science Department. ACM CHI 2005 Workshop on Graduate Education in Human-Computer Interaction. Organized by Beaudouin-Lafon, M., Foley, J., Grudin, J., Hudson, S., Hollan, J., Olson, J. and Verplank, B. April.

My interest and experience lies in this middle ground. In this paper, I discuss the HCI program as created within the Computer Science program at the University of Calgary. While the program is successfully training HCI graduates, it is fraught with issues and workarounds that come from trying to fit it into a traditional program. In this paper, I articulate some of these issues, not because they are unique, but so that others in similar circumstances can compare their own issues and workarounds to ours.

THE CALGARY HCI PROGRAM
First, I will briefly describe the HCI program at Calgary at both the undergraduate and graduate level.

Up until a few years ago, I was the only HCI specialist on faculty. A specialist in Information Visualization (who bridges HCI and graphics) joined our faculty a few years ago, while a third HCI person joined this year. Graduate students interested in specializing in HCI or its sub-disciplines (e.g., CSCW, Information Visualization, Context-aware computing) typically work with one of these three professors, all who share a large common laboratory called the Interactions Laboratory. About 20 to 24 graduates inhabit this laboratory.

At the undergraduate level, the department offers several sections of an introductory HCI course at the junior/senior level. While it is an optional course, it is taken by the

majority of undergraduates. The department also offers an 'advanced' undergraduate HCI course based on the idea of a design studio: students are exposed to several state-of-the-art interface genres (e.g., groupware, tangible interfaces) and are expected to design and implement systems within each genre. The class is restricted to 15 students, who apply for it on a competitive basis. While there are no other undergraduate HCI courses, students can add to their expertise by taking a slate of graphics courses.

At the graduate level, the department offers a fairly traditional MSc and PhD graduate program.

- It is thesis based
- Courses taken are supposed to be a mix of breadth and depth (4 courses for the MSc, and an additional 4 courses for the PhD).
- It favors admission of graduate students with a Computer Science/Engineering background.

There are several graduate course offerings in HCI, but they are not necessarily offered every year. These are described below.

- *CPSC 681. Research Methods in HCI* is an applied survey of evaluation methodologies. It is the most long-standing HCI graduate course in our program, and has been offered in one form or another (sometimes on alternating years) since the very early 1980s.
- *CPSC 781. Advanced Topics in HCI* is a vehicle for teaching a particular advanced HCI topic in depth. The topic may change year by year. Example past topics include CSCW, Tangible User Interfaces, and Heuristic Evaluation.
- *CPSC 683. Information Visualization* covers the theory and development of interactive visual representations of abstract data for the purpose of amplifying cognition. This course was recently introduced due to the arrival of a new faculty member.
- *CPSC 601.XX Special Topics in Computer Science* is a designation for one-off courses (usually a reading course) tailored for a very small group of students with a narrow research focus.

ISSUES AND SOLUTIONS

This section identifies a variety of issues that we continually face in teaching HCI at the graduate level.

Issue 1. Unprepared Incoming Students

Graduate applicants interested in HCI may have no formal training in it. This usually arises because HCI may not have been available at the student's undergraduate institution, or was given as an option that, for one reason or another, did not fit into the student's schedule. The problem is that there is no 'introductory' course on HCI available at the graduate level. Yet students are reluctant to take the undergraduate course because they cannot count it as credit towards their degree. As a result, these students often have to learn the core material on their own.

A related problem is that we often have many graduate students who want some rudimentary training in HCI, even though they do not want to be HCI specialists. This typically arises because these students see HCI as relevant to their research even though it may not be a primary focus. Again, because there is no introductory graduate course in HCI, they either end up taking one of our 'advanced' HCI courses (which may not be appropriate for what they want) or do without.

Solution 1. An Introductory HCI Graduate Course

The obvious solution to this problem is to offer an introductory graduate course in HCI. However, this proved no easy matter. First, there is limited faculty available to teach HCI, and (at least in our department) usually only one graduate course is included in a professor's normal teaching load. This introduces the dilemma that offering an introductory course may mean that the specialist HCI course would not be offered. Second, I originally worked around this problem by offering a one-week (full days) intensive introduction to HCI extra to my load. Over time, this option was dropped simply because it was too hard to schedule and difficult to sustain.

The solution that we are working on now is to get departmental buy-in on the importance of an introductory HCI graduate course, and to have the department guarantee that this course should not compete with other HCI course offerings. We successfully argued the case by noting that a) HCI is important to non-HCI specialists in terms of their breadth training, and b) it is critical to the training of software engineers. Consequently, this course will be offered in the coming years.

Issue 2. Students from Other Disciplines

HCI attracts students from other disciplines. Since HCI is fundamentally a cross-discipline area, we should include these students into our program. Not only would they receive training, but they would add richness and alternate perspectives as they work side by side with the computer science HCI students. Yet our program runs within Computer Science. As our department grows in size (170 grad students), admission rules are becoming inflexible, where they increasingly favor admission of computer scientists over those from other non-technical disciplines. While there is a means to admit these students, this comes at the cost of either an onerous course load or by somehow creating a cross-department multi-disciplinary degree. Thus while both HCI and other faculty favour multi-discipline students, the bottom line argument is that we are still granting Computer Science degrees. As a consequence, the hurdles are just too high.

Solution 2. Cross-Discipline Courses & Collaborations.

The obvious solution would be to create a new degree designation (say, MSc in HCI), i.e., a new program or concentration that touted itself as a cross discipline program. Yet this proved impractical. First, the resources

simply were not there to create and maintain such a program, as it would require new faculty and new program design. Second, it is very difficult to get the University of Calgary to designate new degrees, for these demand government approval.

While we were stuck with admitting into the program only those students with a technical background (excepting exceptional cases where we do try to craft some kind of special cross discipline program), we did not want to lose the richness of cross-discipline collaboration. What we do instead is encourage students to apply to their home discipline (Psychology, Industrial Design, Communications, Educational Technology), and then to take courses from us. That is, while our program could not be cross-discipline, we designed our HCI graduate courses so they can be taken by non-computer scientists. This works well in practice. In CPSC 681, for example, we typically get students from Psychology intermixing with computer scientists within the course and on joint projects. In CPSC 683 Information Visualization, we have had students from Communications, and from a nearby Arts college work on joint projects, many of which were exhibited in at a museum.

A major benefit of including students from other disciplines into these courses is that course projects often turned into first class research projects that went beyond the scope of the class. Students started working together, regardless of the discipline. Because of their abilities, I often hired some of these 'outside' students as research assistants. They identified themselves as members of our laboratory, and enriched our culture of HCI education.

Issue 3. Breadth versus Depth

Our program expects students to take courses that exhibit both breadth and depth in Computer Science. While a recommendation at the MSc level, it is codified at the PhD level into a certain number of courses from predefined areas (e.g., theory, systems, applications). This can leave HCI students at a disadvantage. For example, other specializations in Computer Science expect a minimal level of 'core' compulsory training coming out of the undergraduate degree, e.g., those interested in graduate work in theory would likely have quite a few theory courses under their belt, usually a combination of several compulsory courses and a few optional courses. Yet, as mentioned in Issue 1, HCI students are often lucky if they have a single HCI course before admission. Thus they require a good number of HCI courses to bring them up to the level expected of an HCI professional. This means they can easily fall awry of the breadth requirement, or they cannot take or count some of these desired courses as part of their load.

Solution 3. Designing Flexibility into the Breadth.

Our solution was to add flexibility to the definition of breadth as required by our graduate program, especially at the PhD level. I was recently made Graduate Director of Computer Science, and as part of this I was asked to redesign the depth/breadth requirement of our PhD program. This did not mean I could relax it; in fact, my mandate was to make it stricter than it was in order to stay aligned with requirements of other Universities. The original proposal (handed over from a previous year) was a fairly standard requirement that students must take two courses within each area of theory, systems and applications.

The solution was to add flexibility to the breadth requirement. First, a fourth area 'External to Computer Science' was added to supplement the three core Computer Science areas. This meant that students could take courses related to HCI from other disciplines (e.g., Psychology Human Factors, Industrial Design) and have them count towards their breadth. Second, we added a caveat that would let students deviate from these hard rules if it could be shown that this was in their best interests: "However, in particular cases, course programs for PhD students can deviate from the above by designing and justifying an alternative breadth/depth program that satisfies the supervisor, the supervisory committee, and the graduate committee." I should add that these solutions also solved concerns raised by other faculty members who needed a greater depth component than that allowed by the original program description.

Issue 4. Course Availability

As mentioned above, several courses are offered in HCI on an irregular basis. This is proving problematic, for incoming students needing core HCI expertise (such as evaluation methods) may not be able to take it until the second year of the program. This is simply too late. Again, students in this situation are expected to pick it up on their own, or to have other students mentor them.

Solution 4. A Graduate HCI Concentration

Our solution, which has not yet been implemented, is to design an HCI concentration for the graduate program. A concentration is a semi-formal program. While students can enroll in a concentration, it is really little more than a recommended set of courses. The trick is to get these courses approved by the department, and to have the department guarantee (as much as possible) that a certain slate of courses would be offered every year.

In our particular situation, we would like to guarantee the following two course offerings every year: Advanced Introduction to HCI, Research Methods in HCI. Each year, we would also guarantee a course offering of at least one 'specialist' HCI course, in the form of the Advanced Topics in HCI e.g., Computer Supported Cooperative Work, Information Visualization, Context-Aware Computing. Interspersed would be the reading courses, given on a discretionary basis. This means that an incoming student

would be able to take two or three HCI courses in the first year.

Another option that we encourage is for students to look for HCI-related courses outside the Computer Science Program (See Solution 4). We have found several good courses, e.g., a Sketching and a Qualitative Evaluation course in Industrial Design, a Human Error and an Industrial Ergonomics course in Psychology, and several others. The challenge is to get our students admitted into these courses without the necessary pre-requisites. We do this in several ways. First, the HCI faculty talks to these course professors about the relations between HCI and their course material. Second, we invite students of those professors to join our courses. Third, we seed the process with an 'exemplar student' to prove that Computer Science students can not only do well in those courses, but that they can also add valuable insights to the class discussions and projects.

When the concentration is in place, we expect it to be a mix of recommended courses both inside and outside of Computer Science. Ideally, we would like other faculties to create their own concentrations that include our courses. By doing so, we will have created a grass-roots interdisciplinary program, which will provide another solution to Issue 2.

OTHER CONCERNS

There are several other concerns arising from graduate education of HCI students within Computer Science. I list them here in no particular order, and just raise them as possible discussion points.

HCI as a technical field. As computer scientists, our students can contribute much to the technical aspects of interface design. Yet, in practice, our HCI courses tend to concentrate on HCI material gleaned from other disciplines, as these will be the areas that students will be least familiar with. While we demand students do technical aspects of HCI as part of their research, we really should provide them with a technical course. This could include (say) algorithms for advanced input techniques, interface toolkit design, interface architectures, interface aspects of distributed systems, and so on.

Toolkits for rapid prototyping. One of the best ways students learn is by doing, where they rapidly prototype and modify novel interface designs. Yet most commercial systems offer tools for only 'mundane' GUI design. To solve this, our laboratory has a toolkit culture, where students package interface methods with a well defined API so that other students can build atop of them. In practice, this has been tremendously successful.

HCI teaching modules. There is, as yet, no single recipe for teaching HCI that will fit all faculty and/or students. One solution is to recognize these by creating HCI modules on specific topics, where modules can be combined in different ways to create courses. I have done this over the many years I have been teaching HCI. My material has been made available over the web and has been used by countless others (www.cpsc.ucalgary.ca/~saul/hci_topics). Others have also attempted this (e.g., Shneiderman collects topics and relates it to his book). The current effort by Georgia Tech to create a more universal repository in its HCC Education Digital Library should help significantly in this regard by creating an HCI Commons that does not reflect an individual perspective or that is not tied to a commercial venture.

HCI teaching resources. A great many resources exist that can considerably assist in the teaching of HCI. As a community, we should collect and disseminate these resources. The teaching modules mentioned above is one example. Tested and well documented interface toolkits for innovative interface design is another example, e.g., as done by ourselves (SDGToolkit, Groupkit, Grouplab DiamondTouch Toolkit, the Collabrary, Phidgets) and Ben Bederson (his Piccolo toolkit. Yet another example would be videos of interfaces. While many are previously published, they are very hard to acquire in practice. The Open Video project is one example of a university attempting to collect and disseminate this type of material.

CONCLUSIONS

As HCI matures, we as a community are anticipating specialized HCI graduate programs for training highly qualified HCI personnel. While a handful of programs now exist that do this, we should not forget that the vast majority of universities are only just hiring a single HCI faculty member to teach HCI within a traditional program, and that most students are still coming through these traditional graduate programs. Consequently, I believe it is important for the educational HCI community to exchange issues, tradeoffs and workarounds that HCI faculty in these traditional programs have developed over time. While programs with established HCI faculty already know how to do this, the many new faculty members that are being hired may use this information to fast-track a workable HCI program within their traditional department.

ACKNOWLEDGMENTS

When I was a graduate student, David Hill was a profound influence on me - I attended his HCI course in 1980, and he sent me to the very first CHI conference. Later, I was fortunate to be hired at an institution with a long record of HCI education stemming back to the 1970s. Calgary was always supportive of HCI; I never had to 'fight' for it. Over time, I have learned and built upon the work of my colleagues; HCI is a wonderful field where professionals encourage sharing. Finally, I am grateful to Sheelagh Carpendale, who brought her insights as both an artist and computer scientist faculty member to our training of HCI students.