Report from the trenches:

Bringing more women to the study of computer science

Elizabeth Jessup, Tamara Sumner, Lecia Barker*
Department of Computer Science, *Alliance for Technology, Learning, and Society
University of Colorado
Boulder, CO 80309-0430
Phone: 303 492 0211
Fax: 303 492 2844
jessup@cs.colorado.edu
sumner@colorado.edu
barkerl@colorado.edu

KEYWORDS: design-based learning, educational pipeline, information technology education, user-centered design, women and technology

ABSTRACT

Many social scientists conduct research on increasing the participation of women in computing, yet it is computer scientists who must find ways of implementing those findings into concrete actions. Technology for Community is an undergraduate computer science course taught at the University of Colorado at Boulder in which students work with local community service agencies building computational solutions to problems confronting those agencies. Although few Computer Science majors are female, this course has consistently attracted a very large proportion of female students. Technology for Community enrollment patterns and course curriculum are compared with other computer science courses over a three-year period. All courses that satisfy public markers of design-based learning are seen to have higher than average female enrollment. Design-based learning integrates four practices believed to increase participation of women – authentic learning context, collaborative assessment, knowledge

sharing among students, and the humanizing of technology. Of all the courses marked as including design-based learning, however, the Technology for Community course is drawing the most significant numbers of women from outside of the College of Engineering and Applied Science. We attribute that success to the inclusion in the course of curriculum reflecting design-based learning and recruiting partnerships with programs outside of the College of Engineering.

INTRODUCTION

In spite of the increasing importance of computing and information technology in our society, women are participating in computing disciplines and careers at historically low rates. Although women represent half of all college graduates in the United States, they receive only about 28 percent of all Computer and Information Sciences bachelors degrees, down from 37 percent in 1987 (Barker, Weston, Garvin-Doxas, & Jung, 2003; Camp, 1997; Knapp & Broyles, 2001). Yet even 28 percent is a misleadingly high figure, since when certain computer and information science sub-disciplines such as instructional technology and information science are removed from the analysis, the figure drops to only 22 percent (Barker et al., 2003). Other factors also lower the percentage; for example, the more elite the institution, as judged by the Carnegie Code (i.e., by number and level of graduate degrees it awards) and the larger the graduating class, the fewer computer science degrees awarded to women (Barker & Garvin-Doxas, 2003; Camp, 1997; Cohoon, 2002; Vardi, Finin, & Henderson, 2003). While many social scientists study the phenomenon at all levels of the "pipeline," within institutions of higher education, it is generally up to computer science professors to reverse this trend.

During the summer of 2000, the Department of Computer Science at the University of Colorado at Boulder was invited to participate in the Anita Borg Institute for Women and Technology's Virtual Development Center (VDC) (ABI, 2003). The VDC is a nationwide network of universities pledging to offer technical courses in ways that promote the participation of women. The invitation was attractive to us because, like its peer institutions, our Department of Computer Science has historically experienced low enrollments of women. We accepted the charge to design a "woman-friendly" course, but grappled with what this might mean. This article is our report from the trenches, describing our intervention, a multi-disciplinary course titled Technology for Community (Jessup, 2003) in which students develop computational products designed to serve the needs of local community service organizations. The course pedagogy is rooted in design-based learning, an area in which we had existing teaching and research expertise. Into this we integrated four key approaches derived from the research literature: authentic learning context, collaboration, knowledge sharing, and the humanizing of technology.

This article is organized as follows. We first review relevant research that might support faculty in designing a "woman-friendly" course. We then define the course's design-based learning pedagogy, focusing on the features known to promote the recruitment and retention of women into technical fields. We subsequently describe the Technology for Community course as well as the outcomes of our efforts from the viewpoint of both the faculty and students involved. In the latter, we are helped by social scientists involved in a National Science Foundation funded Information Technology Workforce research project. We proceed with a discussion of implementation challenges facing faculty trying to promote the participation of women in their courses then provide some concluding remarks.

Research on Attracting and Retaining Women in Computing

The low participation of women in computing professions stems from both failure to enter and attrition. As a result of both observable and subtle processes, girls begin very early to believe that computing is in the "male domain" (Bryson & de Castell, 1996; Margolis & Fisher, 2002; Sadker & Sadker, 1994). Not surprisingly, by the time they are in middle or high school, girls and boys begin to have strong attitudes about "who does computing," with girls often opting out (Dryburgh, 2000; Kirkpatrick & Cuban, 1998). Girls and women who do take computer science classes in high school are not only a minority, but also confront a male majority who believes they do not belong there, resulting in a male computing culture in higher education in which many women feel they do not or cannot belong (Beyer, Rynes, Perrault, Hay, & Haller., 2003; Dryburgh, 2000).

Of course, many women do enter computer science programs, but nationwide, women leave them at much higher rates than do men (Cohoon, 2001). Several factors combine to account for the greater attrition of women who enter computer science. In introductory computer science courses, female students perceive themselves as struggling more than their more experienced, and therefore better prepared, male peers (Barker & Garvin-Doxas, 2004; Margolis & Fisher, 2002). This situation is exacerbated by inadequate peer-mentoring relationships among female students due to low female enrollment, the small number of female faculty role models, and lack of faculty-student mentoring (Button & Sharrock, 1996; Cohoon, 2002; Haller & Fossum, 1998). Indeed, learning in introductory courses is often individualistic, aligning collaboration with plagiarism (and grounds for failure) (Barker & Garvin-Doxas, 2004). Yet research shows that women prefer collaborative learning environments (Agogino & Linn, 1992; Felder, Felder, Mauney, Hamrin Jr., & Dietz, 1995; Strenta, Rogers, Russell, Matier, & Scott, 1994). What is more, it is well established that all students learn more when they are engaged in two-way communication with the instructor, collaborate with other students, and hear their peers articulate what they are learning (Bransford, Brown, & Cocking, 2000; Cooper, Ayers-Lopez, & Marquis, 1982; Forman & Cazden, 1985). Other research shows that many males have low experience and limited opportunities for mentoring, yet are more willing to "tough it out" (Davis, Gregerman, & Hathaway, 2003). Females, however, express low confidence, which as both Margolis' and Fisher's (2002) and Cohoon's (2002) studies show, precipitates a student's decision to switch major.

The majority of computer science undergraduate programs use a "programming first" approach in the introductory course, leading many students to believe that computer science is equivalent to programming (IEEE/ACM, 2001). Women who take introductory courses as non-majors have "little exposure to the conceptual and intellectual foundations that underlie the revolutionary technological developments driving change throughout society" (ibid, p. 27). In addition, pedagogical methods and assignments within these courses often privilege abstract principles, discussing, for example, programming syntax rather than the human contexts in which an algorithm might be useful. Instruction rarely takes place with genuine users or in contexts of computing application, that is, in real-life rather than contrived applications, until the senior year. Yet research also suggests that women enter computing study in order to apply computing to some human context (e.g., medicine or education) and that they are less likely than men to be interested in the computer as a machine (Haller & Fossum, 1998; Margolis & Fisher, 2002).

Several types of solutions are proposed in the literature to attract and retain women in computing study. These include offering introductory or immigration courses which offer a broader view of the IT industry or which accommodate different abilities at the early stages of the program (c.f., (Gürer & Camp, 1997; Margolis, Fisher, & Miller, 1998; O'Leary, 1999)). Another common solution is to lay the groundwork for student collaboration—benefiting all students, not just women—by requiring pair programming in the introductory course and projects that are less gender-biased (Gürer & Camp, 1997; Margolis & Fisher, 2002; McDowell, Werner, Bullock, & Fernald, 2002; O'Leary, 1999). Ensuring that visible role models are presented to students (e.g., guest speakers), faculty and student role models are available, and improving student-faculty and student-student relationships through mentoring programs and support groups are also recommended strategies (Borg, 1998; Gürer & Camp, 1997; Margolis & Fisher, 2002).

We have borrowed from some of the recommendations above as well as from the more general literature on women's preferences in building our "woman-friendly course." We want to emphasize, however, that 'women-friendly' does not mean that the course we designed is compensatory; i.e. making up for a deficiency in women (Howe, 1993; Muller, 2002). On the contrary, the course we created is founded in the principles of design-based learning and embeds four key features that make up for a deficiency in many courses: authentic learning context, collaborative assessment, knowledge sharing among students, and the humanizing of technology.

APPROACH: DESIGN-BASED LEARNING

As previously discussed, prior research suggests that many women appreciate and derive motivation from understanding, a priori, the broader utility and potential applications for detailed technical knowledge. Design activities readily support this type of contextualized learning due to the deeply intertwined nature of working and knowledge creation during design (Kolodner, Crismond, Gray, Holbrook, & Puntambekar, 1998; Sumner & Stolze, 1996). In his seminal books on design practice and design education, Schon describes an action-breakdown-reflection cycle that underlies professional practice (Schön, 1983, 1987). In this cycle, designers construct design representations until their expectations are not met and they experience a *breakdown* in the current work situation. Breakdowns trigger the designer to explore new ideas, fix problems, seek new information, or consider alternatives. In this way, they present situated opportunities for learning and for the construction of new knowledge (Fischer, 1994). It is when breakdowns occur that the motivation and need to learn are most apparent to the designer. Thus, in design-based learning, skill and knowledge development activities are all situated within an authentic problem-solving context where success is measured by the ability to produce a useful and useable artifact.

Prior studies identify classroom practices that promote collaboration and knowledge sharing as supporting the retention of women in technical disciplines (Barker, Garvin-Doxas, & Roberts, 2005). Knowledge sharing is defined as formal and informal classroom practices that encourage faculty and students to present their ideas to each other as works-in-progress for joint discussion and elaboration. Barker, et al. argue that males often use discourse that functions to over-represent their knowledge; such behavior can reinforce women's perceptions that they are underperforming relative to their male colleagues (ibid). Knowledge sharing interactions help to counteract these perceptions. Collaborative assessment and knowledge sharing are readily incorporated in design-based learning through the inclusion of standard studio practices such as critiquing of works-in-progress. Such practices are routinely used in design practice to overcome the 'symmetry of ignorance' (Rittel & Webber, 1973). Coined by Rittel in the 1970's, this 'symmetry' refers to an enduring aspect of realistic design projects and highlights that the knowledge needed to complete a design is necessarily distributed across interdisciplinary stakeholders.

Finally, as previously mentioned, research on the recruitment and retention of women in computer science has claimed that aspects of the academic engineering environment are often perceived as being hostile or alienating to women. Particularly problematic is the general impersonalization of the learning environment: teachers often do not know students as individuals or even know their names, students often feel isolated from each other, and students do not understand how their work may affect others (Barker & Garvin-Doxas, 2004). Design-based learning can counteract impersonalization and humanize both the classroom setting and the technology development process. Within the Technology for Community course, the projects are more complex than any student can take on individually. Necessarily, all work must be performed by teams of students working together intensely both within and outside of the formal class period. Students within the teams learn to depend on, and count on, each other's unique skills and expertise. Additionally, in line with modern product design best practices, the course activities are deeply rooted in user-centered design methodologies (Lewis & Rieman, 1993; Norman, 1986; Rosson & Carroll, 2002). Such methodologies emphasize deep knowledge of the users and the problem context. Thus, students spend significant time interacting with the project sponsor and other members of the sponsoring organization over the course of the semester, or longer depending upon the project. These external ties serve to deeply personalize and humanize the project as the student teams come to realize the sponsor cares about, and is depending on, the final project outcomes.

In summary, it is important to note that design-based learning courses differ significantly from our "standard" computer science courses in several ways. First, learning is driven largely by the students' own work agendas, as opposed to the teacher-developed course calendar. Second, course time is mostly devoted to actively working on projects, as opposed to passively listening to lectures. Third, most significant student work is collaboratively developed and assessed, as opposed to being the product of students working individually. Fourth, the end product or final course deliverable is collaboratively negotiated amongst the student team, the course instructors, and the community sponsor, as opposed to being pre-assigned, with little give and take along the way. And finally, design-based courses almost necessarily have a human dimension as user-centered design emphasizes understanding the needs and goals of real clients as a primary objective; all artifact construction and other design activities serve this people-oriented goal.

TECHNOLOGY FOR COMMUNITY

Technology for Community satisfies both elective requirements for Computer Science majors and project course requirements for students enrolled in an interdisciplinary, multimedia certificate program called Technology, Arts, and Media (TAM) (TAM, 2003). The TAM certificate program, open to all undergraduates, requires that students take six courses: three that are project-oriented and three that are theoretical in nature. Project-oriented courses feature significant opportunities for hands-on development in project teams. Students acquire in-depth skill with high-end, software packages (e.g., Flash, Photoshop) as well as some HTML programming, with the goal of designing and producing multimedia materials both for self-expression and to serve clients. Programming courses (e.g., C++ or Java) are optional and most students do not take them. Theoretical TAM courses are more focused on history, communication theory, implications of media for society, etc. Enrollment in the TAM certificate program is consistently more than half women. Technology for Community satisfies the "Invention and Practice" requirement of TAM.

There are no prerequisites for Technology for Community, and participants have diverse backgrounds in terms of educational experience, major, and expertise with technology. In the course, students learn to use software tools for such computational tasks as composing and editing websites and maintaining and manipulating databases. Students also study user-centered design methodologies (Norman, 1986) including the process of turning an idea into a useful software product and methods for evaluating that product in terms of usability. The course's design-based learning pedagogy is instantiated in three major activities described below: 1) a Technology Innovation workshop, 2) a short skills-development practicum, and 3) a semester-long project.

Technology Innovation Workshops

Project ideas for Technology for Community are generated in community brainstorming workshops, called Technology Innovation Workshops, held annually and facilitated by Anita Borg Institute staff members. These workshops are the primary mechanism in the course for supporting the involvement of a broad spectrum of women and for ensuring that the course projects reflect the priorities of women.

Participants in the workshops include technical and non-technical women from the Boulder community at large, representatives of local community service agencies, a few interested academics, and the students enrolled in the course. Each workshop takes place over two half-day sessions. At the workshop, attendees visit the question "What services does your community need that it doesn't have now?" Over the course of the workshop, dozens of answers are articulated, ranging from the immediately practical (an internet task/volunteer match system) to the outrageous (a transferable uterus) to the wholly impractical (a conference center).

The interactions through which participants answer this question are carefully managed by a facilitator, who first introduces a technique for listening without judgment. Surprisingly, many students take this technique seriously, returning in the second day with reports of how they used the technique with their friends. This training sets up very early in the semester the value of listening to others and equips students to do so. The facilitator also minimizes social status and hierarchy differences by deliberately not asking participants to reveal their working identities till the end of the second day (e.g., undergraduate Computer Science major, director of a safe house for battered women).

It is during the community workshop that students begin to realize that the community members see them as experts who can help to solve their very real problems. For their part, students report enjoying working with the groups and seeing their classmates in this larger group environment, where they take on roles outside of those of a typical student.

Immediately after the workshop, course instructors work with attendees to identify individuals and organizations willing to sponsor projects over the next academic year. The potentially obtuse answers to the workshop question are reshaped into project ideas. For example, "community-based technology education" became an on-line tutorial for middle school students on using the Internet to do research. These projects are then compiled into a list that students draw from when forming project teams. Over the course of one or more semesters, the students work in groups to bring their solutions to product form with faculty members and community sponsors serving as advisors. A particular strength of the workshops is that they encourage participants to think beyond their workaday concerns: the resulting project ideas have been creative, interesting, and broad in scope.

We note that, for the purposes of this course, "community service organization" is quite loosely defined. Our clients have included a variety of non-profit and governmental organizations that provide aid or education of some kind to the people of Boulder, Colorado. In particular, the Boulder Valley School District has been an important participant, as have Boulder Senior Services (City of Boulder, 2004), the Conference on World Affairs (CWA, 2004), and Thorne Ecological Institute (Thorne Ecological Institute, 2004).

Skills-Development Practicum: Setting the Stage for Learning from Each Other

In the first three weeks of the semester, students develop and apply their new computational skills by creating "guest books." A guest book is a dynamic website where users can enter or retrieve information stored in a database. The guest book may have any purpose and design that the student chooses. Each guest book is developed independently by one student; students, however, are encouraged to help each other so that everyone can be successful in getting something working. For example, early one semester, a female with web development experience came in late, looked around, recognized that students were working together, and asked a Computer Science major if she could work with him. The two of them used trial and error to work their way through a web-based tutorial, asking each other questions that demonstrated not only what they knew and didn't know but also their reasoning skills. Through this self-imposed exercise, they realized that each had a kind of technical knowledge the other could draw on that was distinct from their own. While this assignment is graded individually, the skills development practicum is a collaboration in which students share their knowledge with and learn from each other. It is not unusual for students to express frustration (sometimes loudly) and for others to help them sort out their difficulties, often based on having just experienced something similar. This early course experience allows students to identify the sources of expertise among their classmates and sets the stage for the establishment of a collaborative environment where it is expected and appropriate that one will not know all the answers and will ask peers for help.

A detailed set of web-based tutorials is provided, enabling students to develop their guest books at their own pace. The guest book assignment is assessed collaboratively at the end of the three-week period based on three criteria – implementation (does it work?), design (does the artifact reflect user-centered design goals?), and demonstration (can the student articulate her design rationale in a well-reasoned manner?). Students must demonstrate their guest books to the rest of the class, and the class votes for the top two projects, with the winners each receiving a large chocolate bar. Thus students hear each other articulate what they have learned and developed in their own language, rather than in the more theoretical language of their professors.

This assignment gives all of the students a common experience, a common vocabulary, and some degree of proficiency with a common set of computational tools. The guest book represents a substantial piece of individual work: for students who enter the course with little computational experience, designing a website, creating the database, and hooking the two parts together is a major accomplishment. In essence, the guest book assignment performs two important functions. First, it is a small practice project, where students can gain confidence in their technical and design skills before starting to work with community clients. Second, it offers an important social experience that enables students from different disciplines to get to know each other and develop comfort learning from one another prior to the formation of the semester-long project teams. Finally, the collaborative assessment creates the expectation that students will put their work on display, manage their ability to do so, and acquire the empathy needed to provide constructive criticism to other students' work.

Semester-Long Projects: Authentic Tasks for an Authentic Learning Context

The remaining weeks are devoted to completing a computational project in collaboration with a local community service agency. Representatives of the agencies meet with the students in order to refine the project ideas generated at the workshop, and they later work with the students to test and evaluate prototypes. To date, all of the projects have been web-based although that characteristic is not a requirement of a course project. Projects can also extend longer than a single semester. Examples of semester-long projects developed in the course include:

- A self-paced introductory tutorial about using Yahoo to search the Internet and send email for Boulder Senior Services;

- Games for simulating the ecology of a local marsh for the Boulder Valley School District, targeting both fourth and eighth graders;
- An on-line events scheduling calendar for use by local non-profit agencies developed for the Boulder Community Foundation, intended to minimize fund-raising conflicts across agencies; and
- A web site to support the information and collaboration needs of Technology for Community students, sponsors, and community members (Figure 1). Participants in the course can easily add materials to the site to support project development.

Figure 1. Student project: interactive course web site, used to support community information needs and team collaboration.

While most of the projects satisfy small scale needs of the client agencies, a local bank executive who is very active in non-profit work expects the events calendar to save local non-profits $100,000 a year by reducing event-scheduling conflicts. The full set of projects is showcased on the course website (TFC, 2002).

Humanizing the Technology Learning With User-Centered Design

In building their projects, students learn and use a user-centered design process; i.e., one where design is driven by the users' needs and capabilities. User-centered design processes are rooted in the principles that design is an inherently iterative, data-driven process, informed by early and continual involvement of end-users and by successive formative evaluations (Gould, Boies, & Lewis, 1991). There are several different flavors of user-centered design methodologies (i.e., task-centered (Lewis & Rieman, 1993), scenario-based (Rosson & Carroll, 2002), and goal-driven (Cooper, 1999)). Each emphasizes a different type of design representation as the central artifact guiding the process, but all draw on a similar suite of design and formative evaluation techniques (e.g., (Lewis, 1982; Nielsen & Molich, 1990; Polson, Lewis, Rieman, & Wharton, 1992; Wood, 1997)). We have experimented with each of these major methodologies over the past three years, but, most often, the methodology used to guide the semester-long projects is a simplified version of task-centered design.

Students begin their design process by interviewing the sponsor and other members of the sponsoring organization, learning about their backgrounds and about what they want the product to do. These people are often the users of the software being designed. We impress upon students that the focus of the project is not about implementation, but about understanding what their users need. When they are first given this part of the assignment, students are somewhat nervous because it is really driven home to them that they know very little about the people and organizations for which they are developing a product. We ask them what they need to know, and students reply "how they envision their solution," "what their facilities are, operating system, etc.," "what software they have," and even "who we are supposed to speak to" and "how involved they will be with us." Thus, they understand the need to interview clients and are able to come up with important interview questions.

From the interviews, the students formulate a list of *tasks*. A task is a statement of something that needs to be accomplished that does not include information about implementation: a sample task for the events scheduling calendar might be "Mary, the events coordinator for the ACME Public Goods organization, wants to change the starting time for an event she posted on the calendar last week." The collection of tasks defines the functionality of the product.

Working from the list of tasks, the students research commercial products and websites, looking for existing materials that relate to their projects. From those products, students borrow features and functions (within the constraints of copyrights and patents). The value of this step is two-fold: borrowing reduces development time and borrowed features that are already familiar to users help to reduce learning curves for the new product. The students then sketch out a detailed design on paper (a "low-fidelity" prototype) and proceed from the sketches to a rough computer implementation (a "high-fidelity" prototype.) See, for example, (Rudd, Stern, & Isensee, 1996) for more information about prototyping.

Students evaluate their high-fidelity prototypes using the thinking aloud technique (Lewis, 1982; Tognazzini, 1992), which is one of the most common techniques used in usability tests. In thinking aloud evaluations, the students observe a user trying to accomplish a task with the prototype. The user is asked to speak aloud all of her or his thoughts about the prototype during the process. Almost without exception, most of the user's thoughts and decisions (and stumblings and errors) catch the students completely by surprise. More than one student group has redesigned a prototype completely from scratch following the first round of thinking aloud evaluations. Responding to the users' actions and opinions typically gives the students a sense of having accomplished a so-called authentic task that was meaningful in terms of real human needs. For example, one group testing a Flash animation intended to teach senior citizens how to search for information using Yahoo learned in the first walkthrough that their own perceptions of "usable and intuitive" were light-years away from those of their user. One problem, for instance, was that their 75-year-old first-time computer user did not understand the difference between an actual usable interface and a simulation intended to teach her how to use an interface. Further, the elderly tester had physical difficulty with double clicking the mouse. The initial prototype required double clicking for all navigation and so had to be redesigned. Later these students reported that, although frustrating, these revelations were one of the most important lessons of the course. Involving real users throughout the process adds human faces and voices to the technologies the students develop. The steps of prototyping, evaluating, and correcting the projects are the building blocks of the iterative phases of task-centered design. Students repeat these steps until they and the sponsors are satisfied with the results.

The finale of the course is an evening devoted to project team presentations and demonstrations, where students showcase their work to each other, the clients, and the community-at-large. In their presentations, students show an explicit awareness of their disciplinary differences and the value of collaboration. One student joked, "you can tell I'm a computer science major and I made the slide, because it says 'system requirements'." A student reporting on another project introduced one of his team members, saying he "helped a lot with meetings, because he was better at talking to the end user than the rest of us." Having shown their work to clients many times, and having become accustomed to showing their work to each other and getting feedback in class, students are well prepared for this potentially intimidating event. This last activity of the course is an authentic experience that advances students' professional skills.

Course Assessment

Knowledge sharing, collaborative assessment, and peer evaluation are carried out formally and informally throughout the course. Guest books and project milestone results are discussed and evaluated by all members of the class as well as by the instructor. Peer evaluation also plays a role in the assignment of final grades. Those grades are assigned as about 85% work on the project and 15% on the guest book. The project grade is determined in part by the instructor (30% of total), in part by the client (10%), but mainly by the project group members themselves (45%). By means of an evaluation form that poses several questions, group members are asked to evaluate themselves and their teammates with respect to both effort and results. In reality, a well-functioning group with an ecstatically happy client can expect high grades regardless of the other factors. It has not proven necessary to distinguish students according to major, but freshmen and sophomores are graded by the instructor on a somewhat more lenient scale than are upperclassmen: upperclassmen are expected to take leadership roles in the projects, while lower classmen are often primarily learners.

RESULTS AND COMPARISON TO OTHER COURSES

Here, we consider in what ways the Technology for Community course is meeting its objectives to promote the participation of women in technology design and development and to increase the number of women enrolled in computer science courses and degree programs. To assess the impact of the course, we examine enrollment patterns from Spring 2001 through Fall 2003 (five offerings over six semesters) across all Computer Science courses. We also analyze project sponsorship patterns as well as feedback from course participants. To better understand whether the impact is different across curricular models and course types, we compare data from the Technology for Community course with those for other Department of Computer Science courses.

Enrollment Patterns

Between 2001 and 2003, a fairly steady 51 percent of the undergraduate students who have enrolled in Technology for Community have been female. This is a remarkable percentage in the Department of Computer Science where females have totaled between 11 and 16% of undergraduate majors during the last 14 years and have averaged 13% during this time period. The student body in the course is also diverse in terms of major: 44% of the 62 students have come from the College of Engineering, which includes the Department of Computer Science, and 53% from three other Colleges on campus (Arts and Sciences, Business, and Journalism and Communication). Thirty eight percent have been Computer Science majors. Three percent of the students have not yet declared a major.

Public Markers

The high level of participation of women in the course indicates that Technology for Community is providing a type of technical education that is appealing to women. To identify features of the course that might contribute to its success, we compared it to other similar courses within the Department of Computer Science. We defined similarity in terms of three 'markers' that are readily visible to students making course enrollment choices:

1. The course description emphasizes learning about design theory and methodologies, i.e., the cognitive and social models that underlie design activities and the processes for doing design, such as task-centered design or scenario-based design.

2. The course description emphasizes developing externally sponsored group projects as a primary course activity.

3. The course is cross-listed with a department outside of the College of Engineering and Applied Science (in addition to being as a Computer Science course), or the course is listed as satisfying a degree or certificate requirement outside of the College.

The first two markers are visible signs of a course's emphasis on a design-based learning pedagogy. For our purposes, we define a design-based learning course to be any course that satisfies at least one of these two markers. The third marker signals a course's commitment to a broader perspective on computer science beyond traditional disciplinary boundaries. Previous studies have also identified a broader view of technology as important (Gürer & Camp, 1997; O'Leary, 1999).

As shown in Table 1, there are four elective classes offered by the department that satisfy these markers. Educational Technology House covers some aspects of user-centered design and enables teams of students to construct educational technology for faculty sponsors from across the campus. User Interface Design teaches students about the processes, skills, and techniques of user-centered design; students apply their skills in the context of semester-long externally sponsored design projects. Design, Learning and Collaboration combines lectures on design theories and methodologies spanning computer science, architecture, and urban planning with hands-on, team-based project work. A few other Computer Science courses exhibit one of these markers; most have none of them.

All of these courses have a larger than average participation of women. In Table 1, enrollment is reported for both undergraduates and graduates since several of the courses are listed with both undergraduate and graduate sections and because graduates can also enroll in undergraduate courses. These four design-based learning courses all have no course prerequisites, though some require permission of the instructor in order to enroll. All are cross-listed with departments or programs that emphasize understanding the role of technology in human society, cognition, and quality of life. All four courses are taught by faculty members from the Department of Computer Science, two by women faculty and the other two by men.

Table 1. Enrollment patterns in design-based learning courses

	Percent Women		Total #	Course
	Undergraduates	Graduates	Students	Cross-listing
All Computer Science Courses	13%	25%	-------	-------
Technology for the Community	51%	33%	62	TAM
Educational Technology House	36%	-------	33	TAM
User Interface Design	35%	41%	84	ICS
Design, Learning and Collaboration	20%	50%	9	TAM & ICS

Key: TAM = Technology, Arts, and Media certificate program; ICS = Institute of Cognitive Science certificate program

The third marker pertains to cross-listing of the course in a department or program with a greater participation of women than in the Computer Science Department. The first three courses are cross-listed with the TAM certificate program, which presently has a 56% female enrollment. The fourth is cross-listed with the Institute of Cognitive Science (ICS) certificate program. That certificate is awarded for the completion of a series of courses in the study of how human knowledge is acquired, stored, and represented in the mind. The ICS program has a 42% female enrollment. As shown in Table 2, two of the four courses have notable percentages of women from outside of the College of Engineering and Applied Science according to their declared majors. Note that a typical Computer Science course includes no women from outside of the college. We see that Technology for Community, which has been particularly successful at attracting non-engineering TAM students, has a very large representation of women from outside of the college.

Table 2. Declared majors of enrolled women

	Percent Women	
	within College	outside College
Technology for the Community	23%	77%
Educational Technology House	92%	8%
User Interface Design	78%	22%
Design, Learning and Collaboration	100%	0%

Attracting New Computer Science Majors

These data suggest that the design-based courses are bringing more women to the study of computer science. Interestingly, in interviews intended to elicit ways of improving the major, a female student not enrolled in Technology for Community said

> You might wanna look into why there's a lot more women in my computer design class in the environmental design school. I don't really wanna generalize too much, but there might be more leniency for artistic talents and, graphical, kind of. I mean, the programming there seems to be a tool to do something else.

The Technology for Community course has been particularly successful at attracting a large number of women. However, since the course's inception, there has not been a single case where the course has contributed to recruiting a new woman into the Department's Bachelor of Science program. This result is disturbing, but probably not surprising. While the course is open to any undergraduate, at any grade level, to date only 18% (11) of the students have been freshmen or sophomores. The remaining 82% are all well underway in their chosen field of study and unlikely to change majors based on their experiences in a single course. Of the 18% of freshmen and sophomores enrolling in the course, all but four have been computer science majors. Thus, the course has not created many opportunities for effective recruiting into the program.

Project Sponsorship Patterns

A broad goal of the course and the VDC is to increase the participation of women in technology design and development. Clearly, having more female students enrolled in the course helps to satisfy this goal. Another mechanism for increasing that participation of women is through project sponsorship. While there are no formal requirements for how much time a project sponsor should spend working with the students on a project, most Technology for Community sponsors meet with their teams several times a semester and invest considerable time and energy into the projects. They are very much major participants in the design process, reflecting the course's emphasis on user-centered design methodologies.

We compared the gender distribution of project sponsors in the Technology for Community and the User Interface Design courses. Both courses are based around externally sponsored, semester-long projects and include user-centered design methods and techniques as explicit components of the course curriculum. Both courses employ the same mechanism for the formation of student project teams: students state preferences for projects from lists compiled by the instructors. The instructors organize project groups according to those interests with an eye on balancing of skills within the groups. The courses differ significantly in how project lists are put together. In the Technology for Community course, the list is the result of the Technology Innovation Workshop. In the User Interface Design course, the list of projects is compiled from responses to a "Call for Proposals" in a widely distributed email solicitation.

As shown in Table 3, most of the project sponsors in the Technology for Community course are female (68%) which is a significantly different sponsorship pattern from that of the User Interface Design course

(46%). Thus, it appears that the Technology Innovation Workshop, which is unique to Technology for Community, is an effective mechanism for increasing the sponsorship of projects by women. The high level of sponsorship by women in Technology for Community is also due to the large number of women working in the community service sector.

Table 3. Gender of project sponsors

	Project Sponsors	
	Total	% Women
Technology for Community	25	68%
User Interface Design	26	46%

What the Students Say

We have student perception data from three sources: an end-of-semester survey administered campus wide in all classes and observations and interviews by the social scientists working on the aforementioned NSF-sponsored project. Every semester, students respond to open-ended questions about aspects of the course as part of the campus-wide Faculty Course Questionnaire process; these questionnaires are administered during the final week of classes every semester. In these questionnaires, the students have reported that the most rewarding feature of Technology for Community is working in multidisciplinary project groups. Computer Science, Fine Arts, and Journalism majors rarely have the opportunity to interact with each other in their usual coursework. Over the course of the semester, students come to recognize the diverse strengths of their project teammates and the variety of contributions those teammates are able to make to the project.

In interviews conducted as part of the NSF-sponsored study, computer science majors were asked what they would suggest to the Department to improve the major. Among other things, students indicated many of the features of the Technology for Community class. Students proposed making assignments more realistic, interesting, and involving the kinds of human interactions they will have in their careers. For example, one student said

> Maybe make the programs in there interesting. Not just the same thing. I know it's just a tool to teach you, but maybe take a little piece of a real-world kind of program, and make it so you can really see how you're applying it. Make it more into application and maybe not just learning how, 'cause that's how I learn better is to take something and really apply it. I guess you're still applying it if you're writing the program, but you're just applying it to like counting the words in a document. That's just not very exciting.

Another said,

> Put more of a human face to it…When you are in a comp sci class it's, literally, you're focused on the code, on the code, on the code. And I don't think you really get an idea for what people are really doing out there. You think that as a programmer, you're gonna go out there and you're gonna sit and write code eight hours a day. And that's not necessarily the truth. You're still going to have to operate in a business environment. You're still going to have to attend meetings. Just give them more of a realistic idea of what being out there in the business world is.

A student interviewed specifically about the Technology for Community course praised the authenticity of the experience:

> It was the opportunity to actually do a real-world scenario, developing or helping a group with a problem and solving it. It's not like you're in a classroom and you're given this, "Here you go, here's your assignment. Complete it by this date. Get it done." This was

> more of a real world application and none of the other courses really offer that kind of experience.

In fact, students said that the goal of providing good quality products to the clients was inspiring, and they were grateful that competition between students was not used to motivate their efforts. Further, students felt a sense of accountability to their clients and worked harder to produce something worthwhile:

> We have a vested interest *because* this isn't just a group paper we're writing and handing in at the semester. We're making it for kids and we care that it *is really good.* And, you know, we know that their school is going through a hard time. [Student's name] and I were just talking last night. Kind of had an e-mail, phone conference, all three of us, and we have one last content section to do and she said, "You know, this has got to be really good because this is the thing they're probably going to use most, so we've *really* got to spend time going through step-by-step." Whereas, maybe if we were doing just a group project, no one's actually going to use what we would've done. Just slap it up there, you know.

As mentioned earlier, Technology for Community students report that the biggest challenge of the course is learning to deliver a usable technical product to someone with little or no technical background. They typically become acutely aware of this challenge during their first round of thinking aloud evaluations, when they discover that their first design was perhaps not a good as they thought it was. In one end-of-semester presentation to the community, a student that had been involved with the senior citizen group said, "in the first walkthrough, we thought it was a disaster, but we realized as we went along that the everyday language we use was not familiar to them." Not only must the features of the product match the abilities of the users, but also the students must find ways to present those features clearly to the users.

Even the liberal arts majors in the course have grown up with some exposure to computers, while not all of their clients and product users have. While these revelations are difficult to deal with, Technology for Community students interviewed expressed appreciation for the authentic, real-world experience of working with actual clients, as opposed to classes where the experiences were "prepackaged." They found surprising the varied range of clients' motivations in working with computers and the range of computer experience and interest in middle school students. For most, successfully finding language that allows the students of different majors to communicate among themselves and with their community service partners represents the biggest accomplishment of the course.

We find it very interesting that students' comments, for the most part, focus on the 'people' aspects of the course as being the most rewarding, enlightening, and challenging; i.e., learning how to collaborate and communicate effectively with people from different backgrounds. In contrast, typical comments from standard Computer Science courses focus on the course mechanisms and technical course content. Universities within the VDC network are trying to get a handle on how participating in VDC programs affects students' perceptions of technology. The student comments imply that one way the Technology for Community course may influence students' perception of technology is to humanize technology design by highlighting the value that people bring to the process.

Interestingly, students who were not computer science majors reported in interviews that the Technology for Community course was like an introductory computer science course, and they believed that they were exposed to "real" computer science. Unlike a more typical introductory course, however, these students felt that the computer science majors and their computer science professors gave them great encouragement and enabled them to be successful technically.

DISCUSSION AND IMPLEMENTATION CHALLENGES

The study presented here can, at best, provide suggestions for concrete action toward increasing the number of women involved in computer science. Specifically, the small number of students overall precludes a definitive statistical analysis of observed trends. Furthermore, it is hard to decouple the effects of the various factors influencing the participation of women: the four courses that we examine all demonstrate at least two of the significant markers, and other external factors may further influence the situation. Although much of the observation and interview data collected in this study are consistent with the Carnegie Mellon study reported in *Unlocking the Clubhouse* (Margolis & Fisher, 2002), a more comprehensive study including several universities might help to identify more critical factors. Nonetheless, various strategies for attracting women have emerged.

The large percentage of women in design-based courses points to the appeal of that pedagogy (Agogino & Linn, 1992). From the perspective of faculty members located in a College of Engineering, the task of creating a 'women-friendly' curriculum can seem quite daunting. The essential elements of such a curriculum are difficult to identify from the research literature. Studies tend to discuss particular practices in isolation; e.g., women like to collaborate, women prefer to work on projects that are personally meaningful, women students appear to benefit from knowledge sharing activities in the classroom, etc. (Barker et al., 2005; Felder et al., 1995; Gürer & Camp, 1997; Margolis et al., 1998; O'Leary, 1999). Design-based learning offers a coherent pedagogical framework to collectively address these practices. Design-based learning also supports modern engineering educational standards (ABET, 2003). Alignment with recognized standards can be very important when introducing new courses into an already full curriculum, as one of the challenges is to match course content with the goals of the Department and the College.

Once the content is in place, it is necessary to communicate it to the right group of people. That is, the three course markers must be made public. An excellent course announcement advertised by good advisors is an effective recruiting tool. We believe that an important factor contributing to the high percentage of women from outside the College of Engineering in the Technology for Community course was the effective advising service provided by the TAM program (Biros, 2002): its advisors actively promoted the course as fulfilling the certificate's project course requirement. Another valuable recruiting tool may be to have Technology for Community students present their work and, especially, their work processes and feelings about these processes to students taking non-major introductory courses.

An example of the importance of labeling is provided by one course not included in Table 1 entitled "Things that Think." This course has been offered during three semesters of the 2001–2003 timeframe being considered. At first blush, the course appears similar in kind to the other four: students work in teams to design and construct robots and automata that integrate physical and computational components, and it has no prerequisites. However, this course does not satisfy any of our public markers identifying design-based learning or a commitment to a broader perspective. That is, the course announcement emphasizes the creation of artifacts rather than the design processes to be employed; projects in the course are not externally sponsored; and the course is not linked to any other departments or programs. In addition, as the title states, this course is concerned with 'things' rather than people, so it seems to lack the human aspect of the other offerings.

In her book *Feminism Confronts Technology*, Judy Wajcman argues that a feminist view of technology would recognize three layers of meaning: what people know about interacting with technology (using it, repairing it, etc.), people's practices surrounding the development and use of technologies, and a collection of physical objects. We know from experience that "Things that Think" is an engaging and innovative course, yet interestingly, the title points only to the latter of Wajcman's layers, the physical objects. It is interesting to note that the course has not attracted women participants substantially beyond the average percentages with only 17 percent female undergraduates and 16 percent female graduates. These numbers are far below those of the four explicitly design-based courses. We propose that deficit occurs because the advance information available to students does not explicitly include the human practices and knowledge building present in the other courses' titles and descriptions.

A final factor in raising the numbers of women involved is the inclusion of infrastructure that directly promotes their participation. While, as instructors, we focus primarily on the student body, including women as project sponsors is another way to garner women's participation. Technology for Community has had particular success in bringing women into the technical process in part because of its Technology Innovation Workshops. We know that the direct identification of that course with the Anita Borg Institute attracts female community participants, and that link may also be a further draw for female students.

We close by examining the effect of these strategies as applied to Technology for Community on the number of women in the IT pipeline. That course has, so far, not caused any non-technical women to move into the Computer Science major. Rather, Technology for Community succeeds in bringing women into the field of computer science at a higher level than they've previously experienced. Many women who participate in the Technology Innovation workshops have virtually no technical experience. Nonetheless they play an important role in developing the projects for the course. Indeed, their importance in that task belies the trepidation that many of them express at their invitation to the workshop. "I know nothing about computers, so I can't possibly help" is not an unusual first response. Some of those non-technical workshop attendees actually go on to be project sponsors and so immerse themselves in a sophisticated project for a semester or longer.

Many of the students are similarly promoted in the technical world. Most students successfully complete the guest book, the development of which is certainly a substantial achievement for the non-technical student. Even computer science majors can be challenged by that assignment as it involves technologies that are not necessarily part of the mainstream curriculum. In fact, we have observed TAM students give computer science students both design and technical tips for accomplishing that assignment. Furthermore, many students in the course find that they need to learn new technical skills in order to complete the community service project, and the structure of the course requires that students learn those skills on their own.

It is worth noting that Technology for Community also provides a positive impact on the men of the Engineering College. For many of them, this course is the only one in which they have the opportunity to work with so many women as both project partners and project sponsors. At least some of these men will pursue careers in areas without the great gender imbalance of the Computer Science major, so for them, this academic experience is a good introduction to a more realistic working environment.

CONCLUSIONS

Technology for Community has succeeded in its goal of bringing more women into the world of computer science. The enrollment of female students is large, and female community members are active participants. We believe that some features of this course can serve as models for faculty trying to create other courses that are appealing to women.

In particular, a design-based curriculum provides a coherent framework for such a course. In a sense, it is an all-in-one package, providing academic legitimacy and pedagogic coherency delivered in a 'women-friendly' style. Technology for Community also demonstrates the importance of emphasizing a human aspect to course content: students find it very satisfying to do good work for others. It is an authentic task in which students truly share their knowledge with each other and learn to value their peers' critiques.

Further, the course illustrates the value of partnerships with external organizations capable of attracting and recruiting women. Our partnerships with the TAM program and the Anita Borg Institute enabled us, as engineering instructors, to focus on the part of the solution we are best equipped to address: course design and teaching. We believe external partnerships are particularly important for individual faculty members trying to positively impact the participation of women in IT education while simultaneously attending to other teaching, research, and service duties.

Technology for Community is thus the result of a good mixture of nontraditional academics and broad institutional support. The result of this combination is a true melding of technology and community, just as the course name implies.

ACKNOWLEDGMENTS

The authors thank Elaine Seymour, Liane Pedersen-Gallegos, and their research group for providing access to their student interview results.

REFERENCES

ABET. (2003). *Accreditation*. Accreditation Board for Engineering and Technology, Inc. Available: http://www.abet.org/accreditation.html [2003, November 23].

ABI. (2003). *Virtual Development Center*. Anita Borg Institute for Women and Technology. Available: http://anitaborg.org/projects/vdc.html [2004, December 18].

Agogino, A., & Linn, M. (1992). Retaining female engineering students: will design experiences help? *NSF Directions, 5*(2), 8-9.

Barker, L., & Garvin-Doxas, K. (2003). *The effect of institutional characteristics on participation of women in computer science bachelors degree programs.* Paper presented at the Innovation and Technology in Computer Science Education, June 30 - July 1 (Thessaloniki, Greece).

Barker, L., & Garvin-Doxas, K. (2004). Making visible the behaviors that influence learning environment: A qualitative exploration of computer science classrooms. *Computer Science Education, 14*(2), 267 - 273.

Barker, L., Weston, T., Garvin-Doxas, K., & Jung, C. (2003). *Computer science programs and gender: The "college of engineering effect" revisited.* Paper presented at the SIGCSE 2003 Technical Symposium on Computer Science Education, February 22 (Reno, NV).

Barker, L. J., Garvin-Doxas, K., & Roberts, E. (2005). *What can computer science learn from a fine arts approach to teaching?* Paper presented at the 35th SIGCSE Technical Symposium on Computer Science Education, February 23-27 (St. Louis, MO).

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller., S. (2003). *Gender differences in computer science students.* Paper presented at the Thirty-fourth SIGCSE Technical Symposium on Computer Science Education, February 19-23 (Reno, NV).

Biros, C. J. (2002). *To Be Or Not To Be: Academic Advising and Mentoring and the Construction of Academic Identity Among Studens in Two Information Technology Programs.* Unpublished PhD Dissertation, University of Colorado at Boulder, Boulder.

Borg, A. (1998). What draws women to and keeps women in computing? In C. C. Selby (Ed.), *Women in science and engineering: Choices for success*. New York: New York Academy of Sciences.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How People Learn: Brain, Mind, Experience, and School* (Expanded edition ed.): National Academy Press.

Bryson, M., & de Castell, S. (1996). Learning to make a difference: Gender, new technologies and in/equity. *Mind, Culture and Activity, 3*(2), 119-135.

Button, G., & Sharrock, W. (1996). Project work: The organization of collaborative design and development in software engineering. *Computer Supported Cooperative Work: The Journal of Collaborative Computing, 5*, 369-386.

Camp, T. (1997). The incredible shrinking pipeline. *Communications of the ACM, 40*(10), 103-110.

City of Boulder. (2004). *Senior Services*. Available: http://www.ci.boulder.co.us/sshhs/ [2004, June 3].

Cohoon, J. M. (2001). Toward improving female retention in the computer science major. *Communications of the ACM, 44*(5), 108-114.

Cohoon, J. M. (2002). Recruiting and retaining women in undergraduate computing majors. *ACM SIGCSE Bulletin, 34*(2), 48-52.

Cooper, A. (1999). *The Inmates are Running the Asylum: Why high-tech products drive us crazy and how to restore the sanity*. Indianapolis: SAMS.

Cooper, C. R., Ayers-Lopez, S., & Marquis, A. (1982). Children's discourse during peer learning in experimental and naturalistic situations. *Discourse Processes, 5*, 177-191.

CWA. (2004). *Conference on World Affairs*. University of Colorado at Boulder. Available: http://www.colorado.edu/cwa/ [2004, December 16].

Davis, C. S., Gregerman, S. R., & Hathaway, R. S. (2003). *Information technology pathways in the academy: Identifying barriers for women and underrepresented minorities.* Paper presented at the ITWF PI, October 25-27 (Albuquerque, NM).

Dryburgh, H. (2000). Underrepresentation of girls and women in computer science, Classification of 1990s research. *Journal of Educational Computing Research, 23*(2), 181-202.

Felder, R. M., Felder, G. N., Mauney, M., Hamrin Jr., C. E., & Dietz, E. J. (1995). A longitudinal study of engineering student performance and retention. III. Gender differences in student performance and attitudes. *Journal of Engineering Education, 84*, 151-174.

Fischer, G. (1994). Turning Breakdowns into Opportunities for Creativity. *Knowledge-Based Systems Journal, 7*(4), 221-232.

Forman, E. A., & Cazden, C. B. (1985). Exploring Vygotskian perspectives in education: the cognitive value of peer interaction. In J. V. Wertsch (Ed.), *Culture, communication, and cognition: Vygotskian perspectives*. New York: Cambridge University Press.

Gould, J. D., Boies, S. J., & Lewis, C. (1991). Making Usable, Useful, Productivity-Enhancing Computer Applications. *Communications of the ACM, 34*(1), 74-85.

Gürer, D., & Camp, T. (1997). *Investigating the Incredible Shrinking Pipeline for Women in Computer Science*. ACM Committee on Women in Computing. Available: http://women.acm.org/documents/finalreport.pdf [2003, November].

Haller, S. M., & Fossum, T. V. (1998). *Retaining women in CS with accessible role models.* Paper presented at the SIGCSE 1998 Technical Symposium on Computer Science Education.

Howe, K. (1993). Equality of educational opportunity and the criterion of equal educational worth. *Studies in Philosophy and Education, 11*, 329-337.

IEEE/ACM. (2001). *Computing Curricula 2001* (December 15, 2001). IEEE and ACM Joint Task Force. Available: http://www.computer.org/education/cc2001/final/index.htm [2004, December 18].

Jessup, E. R. (2003). Teaching Technology for the Community. In G. Gingrich (Ed.), *Managing IT in Government, Business & Communities*. Hershey, PA: IGP/INFOSCI/IRM Press.

Kirkpatrick, H., & Cuban, L. (1998). Should we be worried? What the research says about gender differences in access, use, attitudes, and achievement with computers. *Educational Technology, 38*(4), 56-61.

Knapp, L. G., & Broyles, S. G. (2001). *Postsecondary institutions in the United States: Fall 2000 and degrees and other awards conferred: 1999-2000* (NCES Publication No. 2002156). Washington, D.C: Office of Educational Research and Improvement.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). *Learning by Design from Theory to Practice.* Paper presented at the Third International Conference on the Learning Sciences (ICLS 98), Atlanta, GA (Dec 16-19).

Lewis, C. (1982). *Using the "Thinking-aloud" Method in Cognitive Interface Design* (Research Report RC 9265 (#40713)): IBM Thomas J. Watson Research Center Yorktown Heights, NY 10598.

Lewis, C. H., & Rieman, J. (1993). *Task-centered User Interface Design: A Practical Guide*. Available: http://hcibib.org/tcuid/ [2004, December 16].

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge: MIT Press.

Margolis, J., Fisher, A., & Miller, F. (1998). *Living among the "programming gods": The nexus of confidence and interest for undergraduate women in computer science*. Carnegie Mellon Project on Gender and Computer Science. Available: http://www-2.cs.cmu.edu/~gendergap/confidence.html [2003, November 23].

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). *The effects of pair-programming on performance in an introductory programming class.* Paper presented at the 33rd SIGCSE Technical Symposium on Computer Science Education, Month DY-DY (Northern Kentucky, KY).

Muller, C. B. (2002). *The under representation of women in engineering and related sciences: Pursuing two complementary paths to parity* (Meeting summary http://www.nap.edu/books/0309089603/html/). Washington, D.C: National Academy of Sciences.

Nielsen, J., & Molich, R. (1990). *Heuristic Evaluation of User Interfaces.* Paper presented at the Human Factors in Computing Systems (CHI '90), Seattle, WA.

Norman, D. (1986). *User Centered System Design, New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

O'Leary, D. P. (1999). *Accessibility of Computer Science: A Reflection for Faculty Members* (Version 1). Available: http://www.cs.umd.edu/users/oleary/faculty/whole.html [2003, November 23].

Polson, P., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Communication, 36*, 741-773.

Rittel, H., & Webber, M. M. (1973). Dilemmas in a General Theory of Planning. *Policy Science, 4*, 155-169.

Rosson, M. B., & Carroll, J. (2002). *Usability Engineering: Scenario-based development of human-computer interaction*. San Francisco: Morgan Kaufmann.

Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. High-Fidelity Prototyping Debate. *interactions, 3,* 76-85.

Sadker, M., & Sadker, D. (1994). *Failing at fairness: How America's schools cheat girls*. New York: Scribner.

Schön, D. A. (1983). *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.

Schön, D. A. (1987). *Educating the Reflective Practitioner*. San Francisco, CA: Jossey-Bass Publishers.

Strenta, A. C., Rogers, E., Russell, A., Matier, M., & Scott, J. (1994). Choosing and leaving science in highly selective institutions. *Research in Higher Education, 35*(5), 513-547.

Sumner, T., & Stolze, M. (1996). Integrating Working and Learning: Two Models of Computer Support. *AI & Society, 10*(Special Issue on Hypermedia and Workplace Learning), 70-78.

TAM. (2003). *Technology, Arts, and Media Certificate Program*. Alliance for Technology, Learning, and Society (ATLAS). Available: http://www.colorado.edu/atlas/evaluation/ite/tam.html [2003, November 23].

TFC. (2002). *Technology for Community Course Web Site*. Technology for Community Students. Available: www.cs.colorado.edu/VDC [2003, November 23].

Thorne Ecological Institute. (2004). Available: http://www.thorne-eco.org/index.htm [2004, December 16].

Tognazzini, B. (1992). *Tog on Interface*. Menlo Park: Addison-Wesley.

Vardi, M., Finin, T., & Henderson, T. (2003). 2001-2002 Taulbee survey: survey results show better balance in supply and demand. *Computing Research News, 15*(2), 6-13.

Wood, L. (1997). Semi-structured interviewing for user-centered design. *interactions, 4*(2), 48-61.