White Paper for ICER06

Lecia Barker, ATLAS Assessment and Research, University of Colorado-Boulder

1. ***Preparing undergraduates for computing careers: What are the biggest challenges that you face in your role?***

Much of my work involves research on under-representation in computing study and careers and working with others who are implementing interventions to recruit, retain, or advance students in computing. One of my projects puts me in frequent contact with high school students in a computer magnet program. One day, one of the students was asking me what majors she might choose from for college. This high school senior had successfully taken a wide variety of computing courses and had participated for two years in our special summer camp, teaching middle school kids how to install a network (both the hardware and the software aspects) and how to write software. I asked her if she had considered computer science. The smile left her face, her eyes became hard, and she stated uncompromisingly, “No way. I don’t wanna sit behind a computer all day. I wanna do something fun.”

Preparation goes beyond academics to include a positive attitude and realistic beliefs. If we are to diversify and expand the pool of computing professionals, then high school and undergraduate students need to be open to considering a computing career. They need to know what a computing job might look like on a typical day, in what ways their work can benefit society, typical career paths, what kinds of people a computing professional interacts with and how often, what other kinds of knowledge one will apply (e.g., linguistics for speech recognition), which types of computing jobs are most likely to be outsourced offshore, etc. One of the greatest challenges I face is overcoming students’, teachers’, and parents’ misconceptions about what constitutes a “computing career” and who does computing.

2. ***Transforming the educational experience: What might the community do to address the challenges you identified above?***

Since computer science is only rarely part of the mainstream curriculum in K-12 settings, and because attempting to change educational policy is implausible, it is necessary to recruit majors from high schools and from non-majors on campus. Ideas: 1) Provide recruiters with effective messages and materials for addressing misconceptions. Recruiters should not, for example, assume that girls and college women hate mathematics. 2) Expand our thinking about who does influences choices to include high school mathematics and science teachers, community officials and religious groups, parents, etc. 3) Make sure that these people receive some sort of training to reflect on their own misconceptions and beliefs about the kind of work different people do and how these influence others in very subtle ways (e.g., the college advisor who says, “do you want to take philosophy? The guys like philosophy…”). 4) Integrate computing into the various disciplines where computing is now relevant. For example, in high school biology, teachers could include a section on bio-informatics and the kind of preparation a student needs to participate in this field. Find a way to get this onto standardized tests and it will be taught. 5) Modify undergraduate computing programs of study such that they are responsive to societal needs (that is, make sure that the message the recruiters are using reflect the reality in which students will find themselves for four or five years).

3. ***Models for transforming computing education: What might an ideal undergraduate model for computing education look like in five years?***

At the 2005 Tapia conference, Elaine Weyuker said “the future of computer science is at the interface of computing and other disciplines – not computing, but ‘computing and’.” Given the broad range of computing application areas, it seems that undergraduate computing should provide students with experiences in various areas as well as an understanding of the choices that will face them when they graduate. This could take different forms and have different degrees of impact on students’ perceptions of “the” field. A department might, for example, have a required course titled “common application areas,” where students learn about programming techniques and problems typical to common computing application areas, such as business needs (e.g., supply management systems), defense industry needs (e.g., security), and medical needs (e.g., patient records management. Rather than focus only on the programming aspect of the application, students would be required to learn about the human aspects of these application areas, such as common needs, security, privacy, and ethnical implications. Similarly, studernts could be required to take a minor related to an important application area, as they do at Carnegie Mellon University (though what to do about the frequency with which students minor in mathematics is unclear). Perhaps a stronger model would be to ensure that "computing and" is threaded throughout the curriculum by the nature of assignments and make sure that students go beyond the computing to learn something about the "and." An even stronger model would be to require that majors choose one of so many tracks, in which students will be well prepared to do certain types of computing work, including an understanding of the non-computing nature of the work. For example, Eastern Oregon University has three tracks: a traditional CS major, a statistical and scientific computing (focusing on application development in various sciences), and multimedia (graphics/animation, simulation, telecommunication).

4. ***Inhibitors and strategies: Can you identify inhibitors that might prevent the nation from achieving goals it sets for computing education? Can you identify strategies that may enable the transformation of undergraduate computing education in the USA?***

Supposing that recruiting were effectively increasing the diversity and size of the applicant pool and that model curricula and curricular models were available for adaptation, we still have to convince faculty that they should change their department's curriculum, teach in ways that retain students, and use assignments that are inclusive of different groups. Instituting and sustaining faculty change is key, since faculty are in the classrooms influencing their students in positive or negative ways. Yet this also may be the largest barrier to success. Organizational change of any sort is difficult, but in the culture of independence and autonomy that dominates academe, making sustainable change that goes beyond the department and into the classroom may be a Sisyphean effort.

Faculty work very hard to develop lectures to effectively teach principles which they hold dear and in ways that are comfortable to them. Therefore, faculty will need very good reasons to permanently modify their teaching or assignments. In addition to good reasons, faculty will need professional development and the strength to maintain their methods in the face of student resistance. Beyond preparing present faculty to integrate new curricula, assignments, or pedagogy, some attention should be paid to the professoriate of the future: graduate students.

5. ***Who might participate: What stakeholders should be involved in designing strategies to catalyze the transformation of university computing education throughout the nation? What is the role of government in this process? Professional societies? Universities and faculty? Others?***

The principles of participatory design may be the key to effective reform. Participants should represent those who are likely to be recruited, those who will be recruiting, and those who will educate undergraduates. Educational policy makers might be included: if study of computing is as important as we make it out to be, it should be included in the set of required courses a student takes in high school and appear on college entrance exams.