You Should (and Absolutely Can) Keep Diversity in Sharp Focus During the Enrollment Surge

Lisa C. Kaczmarczyk *(Moderator)*
Evaluation & Assessment Consulting
San Diego, California, USA
lisa@lisakacz.com

Jeff Offutt *(Panelist)*
Computer Science
George Mason University
Fairfax, VA, USA
offutt@gmu.edu

Suzanne Westbrook *(Panelist)*
School of Information
University of Arizona
Tucson, AZ, USA
suzanne.westbrook@arizona.edu

Alvaro Monge *(Panelist)*
Computer Science
California State University Long Beach
Long Beach, California, USA
Alvaro.monge@csulb.edu

Healther Pon-Barry *(Panelist)*
Computer Science
Mount Holyoke College
South Hadley, Massachusetts, USA
ponbarry@mtholyoke.edu

Keywords—Enrollments; Diversity; Pedagogy

I. Summary

Call it a surge, call it a bubble, just don't call it business as usual. It is no secret that enrollments in college and university computer science (CS) classes are growing rapidly and faculty are under pressure to teach more classes and significantly larger classes. They need new and creative ways to accommodate as many students as possible while maintaining excellent pedagogy. Not only that, we don't want to repeat history and worsen already poor diversity statistics with enrollment management strategies that shut out or discourage women and other under-represented groups.

Since the 1980s, when we faced similar burgeoning CS enrollments along with a steep decline in diversity, we have learned a significant amount about what impacts the recruitment and retention of women and other under-represented groups. There is ample research and documentation, readily available, that shows how many well intentioned crowd management strategies, including those that target "the best and the brightest", lead to the exclusion of under-represented groups.

What has not been sufficiently discussed and disseminated however, are effective strategies that directly address the enrollment surge *and* support increased diversity in computer science. In response to this need, this panel will present innovative strategies from four very different schools. Following an overview of the programs, we will have an interactive dialog with the audience. Faculty will have an opportunity to ask questions and discuss the programs in the context of their own teaching situation.

II. Alvaro Monge

Computing departments face challenges and opportunities with the rising enrollments in their programs. There are challenges to the department infrastructure that may no longer be adequate to support all students in the program at the same level of quality. This enrollment growth also comes with opportunities, since, generally department funding is linked to its enrollment.

California State University Long Beach (CSULB) is one of the 23 campuses that form the California State University (CSU) system of public masters-level universities in California. CSULB is a Hispanic Serving Institution; in Fall 2014, 37% of the students were Latino or Mexican American.

At CSULB, the Computer Science (CS) program has grown from 617 students in Fall 2010 to 1,149 in Fall 2014; the growth was reflected in the pre-CS1 course where in Fall 2012 there were 213 students whereas in Fall 2014, there were 438 students enrolled. Unfortunately, digging deeper in the student population of the CS program reveals a loss of diversity. The percentage of Latino/a students has dropped from 26.7% in Fall 2010 to 21.8% in Fall 2014. The percentage of Black/African American students has also dropped, while, the percentage of women has remained about the same, going from 11.3% to 11.8%. The single group that has had a clear surge in enrollment is Asian students, whose enrollment has gone from 26.3% to 44.6%.

The CECS department is working with the National Center for Women and Information Technology (NCWIT) to implement recruitment and retention strategies that can lead to an increase in enrollment, retention and graduation of women

978-1-5090-0151-4/15/$31.00 ©2015 IEEE

and students from underrepresented groups. Strategies include: increased and targeted recruitment at community colleges, scheduling sections of CS1 for students with experience in programming and other sections for students with no such experience, evaluation of the instruction by faculty in these classes, and assigning the best instructors to the introductory courses.

III. Jeff Offutt

Enrollment in introductory CS courses is skyrocketing across the nation. This explosive growth is exposing deep pedagogical flaws that stress our capacity to deliver quality education. The growth is mirrored at George Mason, which is a large (34,000 student) state-funded research university in northern Virginia. In fall 2015, the CS undergraduate enrollment was about 900, double what it was in 2010. Enrollment in CS-1 has more than doubled since 2011 to 484 students in fall 2014. Since most professors are primarily evaluated on their research, they are reluctant to teach courses that impose large time demands, such as CS-1 and CS-2. The combination of rapidly increasing enrollment and focus on research puts enormous stress on our ability to effectively teach first year courses.

The SPARC team at Mason has identified four specific challenges that limit our ability to increase teaching capacity. (1) Students have very diverse backgrounds when they start their computing education. (2) Students learn in different ways. (3) The course evaluations largely focus on syntactic programming skills (declarative knowledge) rather than problem solving skills (procedural knowledge). (4) The same assignments are often used for practice and assessment.

Together, these challenges limit our ability to encourage the kind of collaboration, critical thinking skills, and divergent problem solving abilities that the software industry needs. We characterize this by saying that students are currently taught with an 18th century conveyor belt model, where all students are put in the same boxes and expected to learn the same material at the same rate of speed.

With support from Google's 3X in 3 Years program, a diverse team of George Mason researchers and educators is developing a self-paced learning environment that blends online learning, automated assessment, collaborative practice, and peer-supported learning. Students have collaborative practice assignments and diverse online resources to learn the material. Students proceed through introductory courses at their own pace, earning “belts” by demonstrating skills and knowledge individually at any time, leading to “black belts” that certify mastery of the courses. Advanced and fast learning students can speed through these courses while less advanced and slower learning students can proceed more slowly.

The student educational experience in the SPARC courses is fundamentally different. We separate practice problems from assessments by defining 10 assessment programming problems per course. Each assessment is supported by online educational materials and eight to 10 practice problems. Students view educational materials (videos, tutorials, recorded lectures, etc.) online, then work practice problems. Practice problems are done collaboratively, supporting diversity of backgrounds and knowledge, and offering multiple chances for the courses to be more inclusive. Practice problems can be worked in or outside of class. Students can work as many practice problems as they wish, and when ready, schedule an individual private assessment in the lab. Students are allowed five attempts to pass each assessment.

This model de-emphasizes competition through collaboration and peer-learning. This directly encourages gender diversity and increases under-represented students. The opportunity to retake assessments multiple times encourages resilient problem solving and encourages students to percieve “failure” as a positive opportunity to learn and improve. We borrow the collaborative and peer-learning approaches from martial arts to support self-paced learning and encourage diversity, but do not copy the competitive aspects.

Allowing students to attempt more or fewer practice problems and to take assessments at any time allows the class to be self-paced, increasing retention and diversity. All practice and assessment results are tracked by software, and the instructors are alerted when students struggle to make it through the material. This triggers an intervention, where instructors and TA work individually with struggling students to identify strategies to improve their performance. This intervention technique helps increase retention of students with limited background, from under-represented populations, with poor study skills, and with educational goals that are different from CS majors.

We expect our “black-belt” model to dramatically increase capacity as well as increase retention, especially among women and under-represented groups. This educational model will free our students from the tyranny of the 3-hour semester course by replacing the conveyor belt with a 21st century model of education. If successful, this project will be transferred to educators world-wide, potentially leading to a revolution in early computing education.

IV. Heather Pon-barry

The Computer Science department at Mount Holyoke College has seen a dramatic increase in course enrollments and number of majors over the last several years. The waitlists for the introductory CS course indicate a demand two to three times greater than the current capacity. As of Spring 2015, there are over 80 declared majors. Mount Holyoke College is a liberal arts college for women located in western Massachusetts. Through curricular and co-curricular changes, our program has been successful in attracting and retaining women, students of color, and first-generation college students, populations that have traditionally been underrepresented in computer science.

While a handful of women students arrive with previous programming experience, the majority enrolling in the introductory computer science course have little-to-no prior experience. Because of flexibility in the liberal arts curriculum, students can major or minor in computer science while also majoring in second discipline.

We are currently facing the challenge of how to expand the capacity of the introductory courses without sacrificing pedagogy. Close contact between students and instructors (in small 18-person lab sections) and careful feedback on code are key factors in attracting and retaining students in our program.

The Megas and Gigas Educate (MaGE) program, which is just getting off the ground, addresses this challenge. With support from Google's 3X in 3 Years Capacity Program, MaGE centers around building a vibrant and diverse community of peer mentors as role models. To become a peer mentor, students first participate in a half-semester training course that emphasizes inclusive mentoring and self-reflection on teaching and learning styles. Then, as peer mentors to students in the introductory courses, they provide close interaction in group lab sessions and 1-on-1 student sessions. While research supports the need for cultural-sensitive, inclusive training as part of the educational curriculum, we know of few peer models in computer science that explicitly include this education. Our program includes these elements as we believe this will enhance the model and help inform efforts at other schools.

We aim to increase the participation of underrepresented groups in computer science by fostering a community of diverse and pedagogically-trained peer role models. Students will enroll in an introductory computer course for a variety of reasons; whether they decide to continue is affected by a sense of belonging in the community. In addition to enrollment and retention metrics, we are examining

- Self-efficacy of pursuing higher level coursework and a career in computer science,
- perceptions of investment, and
- communication of recognition of belongingness.

These are important constructs in the literature focused on students, particularly, women, first generation college students, and students of color.

V. Suzanne Westbrook

Like many other post-secondary institutions, student demand for computer science and computing-related courses and degrees has increased dramatically at the University of Arizona, an R1 university with overall enrollment of over 42K including 33K undergraduates. In the degree programs of computer science, information science and technology, and information science and arts we have long used undergraduate teaching assistants called section leaders (SLs) to teach small weekly discussion sections and multiple closed labs of around 20 students associated with large lecture courses. Beyond being an effective coping strategy for changing enrollments without concomitant increases in faculty size, this technique enables us to provide more diverse role models for students than possible with a single instructor and also gives students the positive experience of working with knowledgeable and enthusiastic peer instructors. (The UA programs are not unique – similar programs are in place at Stanford, the University of Washington, Duke, Brown, UT-Austin, and other institutions.)

It's a given that the large lecture format in lower-level college computing courses can be seen as (and often is) an impersonal experience for students and can lead to lower retention rates for all students including underrepresented minorities (URMs) and women. SL programs offer a number of opportunities to alleviate this issue by providing a more personal experience where each student is known by name by at least one SL, more opportunities for interaction with diverse role models, and opportunities to become part of a teaching community (by becoming a SL). In general, SLs are seen by the class students as "cool" and they help alleviate some of the stress of learning computing and programming by making it fun and providing more personal attention and mentoring.

Within our UA SL programs, we intentionally address diversity through hiring, SL training, and weekly SL community meetings. With diversity in mind during our recruiting and hiring process, we recruit new SLs through general announcements to entire classes as well as specifically encouraging individual students who we believe would be great SLs but perhaps won't apply (which is not uncommon for females and other URM students). After the interview process, we intentionally try to hire a diverse group of SLs in regards to gender, ethnicity, age, and disability both so class students have a variety of role models and also to provide leadership opportunities to students who are selected to be SLs. For new SLs each semester we support them with a weekly one hour SL training class covering a variety of teaching topics as well as sessions on diversity issues, accommodating different learning styles, and how to handle difficult students (including a role play exercise that sometimes includes diversity issues, such as an over confident male student being dismissive of a female SL's knowledge). For all SLs (over multiple lower-level classes) there is a combined weekly community meeting to discuss homework assignments that they'll answer questions on during assigned office hours and, just as importantly, to discuss any issues that have arisen that week in either their sections or in office hours. These discussions are often opportunities to discuss how people interact in different ways and how we, as teachers, can respond to certain situations in ways that encourage our students and help them over difficulties.

In addition to discussing the potential benefits of SL programs for increasing diversity in computing, I'd be happy to discuss the challenges and mechanics of this type of program, as well as ideas for a formal research study on the true effectiveness of these types of programs on diversity in computing education.