Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
February, 2013

Ubiquity Symposium

The Science In Computer Science

Broadening CS Enrollments: An interview with Jan Cuny

by Richard Snodgrass

Editor's Introduction

Until 2000, computer science enrollments were steadily increasing. Then suddenly students started turning to other fields; by 2008, enrollments had dropped by 50 percent. Enrollments have started to climb slowly again, but are nowhere near the 2000 levels and far short of projected needs to fill computing related jobs. An additional problem is the very low participation rates of important segments of our society.

Jan Cuny has been leading a program at the National Science Foundation (NSF) to increase both the number and diversity of students in computing. Jan has been at the NSF, serving as the Computer and Information Science and Engineering (CISE) Directorate for the last seven years. She was the founding program director for the Broadening Participation in Computing Program (BPC), an area she has been involved in for many years. That effort has now been integrated into the larger efforts of the CISE Education Workforce Cluster, which she heads. Jan is also the recipient of the 2005 ACM President Award, which recognizes "leaders of IT whose actions and achievements serve as paragons for our field." In this interview with *Ubiquity*, she discusses the magnitude of the problem and the initiatives underway to turn it around.

Richard Snodgrass
Associate Editor

Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
February, 2013

Ubiquity Symposium

The Science In Computer Science

Broadening CS Enrollments: An interview with Jan Cuny

by Richard Snodgrass

Ubiquity: Why is broadening participation important for the discipline of computing?

Jan Cuny: There are lots of reasons. For computing, the underrepresented groups defined by NSF (National Science Foundation) are women, African Americans, Hispanics, Native Americans, and persons with disabilities. That's not a small niche; it's 70 percent of our population. We can't afford to cede their creativity and talents.

It is certainly an economic issue. IT innovation drives our economy, and it has fueled transformational advances in many disciplines. Maintaining the lead in innovation though, will be difficult, given the increasingly competitive global market. We will need our best minds and all of our diverse perspectives. We'll also need many more students in our courses.

For me though, it is also an equity issue. It is unfair to exclude people—whether deliberately or inadvertently—from the many well-paying, creative, and rewarding careers in computing.

Ubiquity: How well is CS doing in terms of meeting the demand for postsecondary degrees?

JC: Poorly. The Bureau of Labor Statistics projects job growth out 10 years. They project that IT job growth will be quite high, higher than all of the other STEM fields put together, in fact.[1] If we combine BLS projections with the number of postsecondary degrees that are being awarded, we find that only two-thirds of these new jobs can be filled. That's a significant shortfall. And it just counts IT jobs. Almost all 21st Century STEM jobs will require some computational skills.

[1] STEM is defined here to include non-medical occupations. Source: Jobs data are calculated from the Bureau of Labor Statistics (BLS), Employment Projections 2010-2020, available at http://www.bls.gov/emp/.

Ubiquity: With the many job openings in computer science, what are the trends in students selecting CS as a major?

JC: Overall college enrollments and degrees in computing fell precipitously starting in 2000, but there are some very hopeful signs of a recovery. Undergraduate enrollments began increasing in 2008, and that was reflected in increases of a little over 10 percent in degree production both in 2009–10 and 2010–11.[2] These increases though aren't enough yet to meet projected demand, and there's still a troubling picture at the high school level. The percent of incoming college freshman who intend to major in computing—which is down 70 percent since 2000—has hardly budged in recent years.[1]

Ubiquity: What were some of the causes of that decline? Is it a systemic problem? Why the uptick recently?

JC: There are undoubtedly many reasons for the decline. Fueled by the dot-com bust and reports of off shoring, many people came to believe that all of the good IT jobs were gone. That's not true. There are more people employed in IT now than there were at the height of the dot-com bubble. We also suffer from many popular misconceptions—the misconception that all computer scientists are geeky nerds with no social skills, that that they work 24/7, and that their work has no societal impact.

Ubiquity: How are we faring compared to other STEM fields?

JC: Not well. We have by far the largest gap between projections for job growth and projections for degree attainment. We also have a horrendous gender gap, arguably the worst of any science. In the 2010–11 Taulbee Survey, women make up just 12.7 percent of undergraduate degree recipients, 29.5 percent of Master's, and 19.8 percent of Ph.D.s. There are other disciplines that have low numbers of women (engineering and physics, for example) but their numbers are holding steady or going up slightly over time. In contrast, our numbers are going down. Computing is the only scientific discipline where the number of women has dropped significantly since the early '80s.

All of the sciences lag in attracting underrepresented minorities, and we do no better. Again, using the Taulbee 2010–11 data across all types of postsecondary degrees awarded in computing, only 13 percent went to African Americans, Hispanics, and Native Americans combined—only 49 of 1,620 Ph.D.s were earned by African Americans or Hispanics.

We can do much better than this.

[2] CRA Taulbee Survey, 2010–2011, http://cra.org/resources/taulbee/.

Ubiquity: I've heard it postulated that one reason that we don't do better with women is that women might be conditioned to avoid difficult technical subjects.

JC: It is true that girls often get messages—maybe from their peers, guidance counselors, or parents—that technical fields are not for them. It is also true that some of those girls who do venture into technical fields find the climate unwelcoming, but it is not true that girls avoid subjects because they are difficult. Girls take AP calculus, statistics, and biology at pretty much the same rate as boys. Girls go on to be lawyers and doctors at similar rates to boys. Why not computing?

Ubiquity: So we need to fix the problem earlier than in college.

JC: Yes. And high school is the most critical point. High school should be the place where students see counters to the popular misconceptions, where they see computing as an empowering discipline that has the potential to impact many aspects of society, where they are introduced to CS role models that look like them. It should be the place where low income and minority students get to experience computing, maybe for the first time. But it's not.

Most U.S. students don't have the opportunity to take even a single academic computing course while in high school. The courses that schools do offer too often cover only keyboarding and the use of office products. We are teaching our students to be consumers of technology, not creators and adapters of technology. On the whole, high school CS courses are so weak that the NCAA doesn't even count them towards sports eligibility.

We need to get more girls and minorities into computing. To do that, and to get all students to see the "beauty and joy of computing," its potential for transformational impact in society, and its relevance to their lives, we need to fix what's happening in high school computing courses.

In seeing all of the great programs we ran with BPC, I came to the conclusion that it's easy to engage kids in elementary and middle school in computing because we have such cool stuff (robots, e-textiles, animation systems, etc.). But that engagement gets lost, if we drop those kids for their four years of high school. Likewise, anything that we do to fix college education will be insufficient if not enough kids are showing up at our doors. High school is a critical point of failure.

Ubiquity: How is computing faring in K-12, both generally and for underrepresented groups, overall and in comparison with other STEM fields?

JC: Over the last 20 years, the percentage of kids taking STEM courses has gone up—dramatically in many cases—in every field *except* computer science. In computer science it has dropped from 25 percent to 19 percent.[3] That's appalling. In 2011, only 21,765 students took the College Board's CS Advanced Placement (AP) test, while 340,551 took AP calculus, 184,497 took AP biology, and 142,910 took statistics.[4] At the same time, only 20 percent of the students taking the CS test were women. That's lower than any other AP test.

Ubiquity: One of the emphases of the CE21 program at NSF is the CS 10K Project. What are the goals of this project? Why those goals; why now?

JC: CE21—the Computing Education for the 21st Century Program—has big goals. It aims to build a robust computing research community, a computationally competent 21st century workforce, and a computationally empowered citizenry. It has three tracks: educational research on the teaching and learning of computing, broadening participation, and CS 10K. With the CS10K track, NSF hopes to lay the foundational support for the CS 10K Project. That project aims to have rigorous, academic computing courses taught in 10,000 high schools by 10,000 well-prepared teachers by 2016. The full project is beyond the mission and resources of NSF, but we hope to catalyze it by funding research and development for curricula, courseware, models of professional development, repositories, and online courses and communities of practice.

Now is the right time for a lot of reasons. There is a national focus on improving STEM, which has led to changes in many states. Many states are moving to graduation requirements that include four years of math and four years science, and so are looking for additional course offerings. Many are also trying to increase the academic content of their CTE (Career and Technical Education) tracks. At the same time, the College Board is developing a brand new AP CS course. And—maybe most importantly—the computing community is stepping up to support this effort on a large scale.

Ubiquity: How does the new AP CS course called "CS Principles" fit in?

JC: The soon-to-be new AP course is critical. First, the existing AP CS course isn't doing us any favors. It's a year of Java programming and it appeals to a very narrow group of students, most of whom have had prior programming experience. That course will continue to be offered but

[3] America's High School Graduates: Results of the 2009 NAEP High School Transcript Study, IES National Center for Education Statistics, 2011.
[4] National 2010 AP Score Distributions, College Board, 2010.

the new "CS Principles" course should appeal to a different and much larger group of students. Second, with the totally decentralized school system in the U. S., AP gives us our only single point of national leverage.

The course is just one of the two that have been recently developed with NSF funding. The other course, called Exploring Computer Science, is a more introductory, possibly pre-AP, course. Both courses were designed to be engaging, challenging, relevant, and inspiring for all students; they were designed from the beginning to be hands-on and inclusive. Both include programming but are not programming-centric. They cover (at different levels) the design of algorithms, issues of complexity and computability, the uses of big data, and the potentially transformative role of technology including the breadth of its applications, its role in enhancing creativity and augmenting human capabilities, and its societal impact. Both courses are designed not just for students intending to major in CS, but for all students.

These courses were piloted this past school year in more than 75 high schools and, in the case of the AP course, more than 20 colleges. A number of university-K12 partnerships have formed to begin both pre-service (in both traditional and alternative certification programs) and in-service professional development efforts. CS 10K awardees trained over 500 teachers this summer.

Ubiquity: What milestones are there between now and rollout of the AP test?

JC: There are two big challenges. The first is to prepare 10,000 teachers. The effort will have to reach both in-service and pre-service teachers and, in a massive effort, provide them with methods courses, face-to-face and online professional development, and ongoing support.

The second challenge is to get into 10,000 schools. To do that state and local policies around computing will have to change. CS will have to count toward core graduation requirements, states will need to adopt K-12 CS standards, and teacher certification and/or endorsements will need to be established. NSF doesn't do policy but ACM, CSTA, NCWIT, and Computing in the Core are all working on this piece.

Ubiquity: If computing is essential to science and engineering in all fields, we could expect to find allies in other fields. Are there allies in other fields speaking up for computing? Are they helping?

JC: Both NSTA (the National Science Teachers Association) and NCTM (the National Council of Teachers of Mathematics) have joined Computing in the Core. We have also begun a partnership with NMSI (the National Math and Science Initiative).

Ubiquity: How can the entire computing community get involved?

JC: There is a role for everyone.

University and college faculty can encourage their departments to offer the CS Principles course—already the University of Washington, Berkeley, Stanford, the University of Texas Austin, and more than 20 other institutions are offering versions of the course. Faculty can also help with training and supporting teachers in their local schools. They can bring their graduate students to the effort and even develop classroom service learning opportunities for their undergraduates. They can also work through their university to get CS put on their list of recommended courses for high school students.

Community colleges are included. They have important roles to play in offering these courses to their students, to providing them to high school students as dual credit courses, and in training teachers.

Teachers can participate in the professional development. They can join an online community of practice and offer their experience and materials. Prepared teachers can also serve as mentors, coaches and trainers for future professional development.

School administrators can offer the courses in their schools, support teachers in their preparations, and make sure that their guidance counselors are helping to steer students in to the courses.

IT professionals can offer their time and experience to help local teachers and they can visit classrooms to highlight careers, dispel the misconceptions about what IT is and who can do it.

Parents, K-12 teachers and administrators, college faculty, and industry can join forces to facilitate the adoption of CS in their schools.

Finally, training 10,000 teachers is a daunting task and it will require substantial resources beyond what CE21 can provide. A Public/Private partnership will be needed to oversee the professional development effort at scale and to raise the needed resources.

About the Author

Richard Snodgrass is a Professor of Computer Science at the University of Arizona, working in the areas of ergalics and temporal databases.

DOI: 10.1145/2435197.2435198