BLOG @ CACM

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/2788449

<http://cacm.acm.org/blogs/blog-cacm>

Plain Talk on Computing Education

Mark Guzdial considers how the variety of learning outcomes and definitions impacts the teaching of computer science.

Mark Guzdial
"The Babble of Computing Education: Diverse Perspectives, Confusing Definitions"
<http://bit.ly/1LFwTbT>

May 22, 2015

Recruiting season for new faculty is drawing to a close, and I felt a twinge of jealousy for non-education fields. A robotics researcher can claim, "My robot is able to do this task faster and more reliable than any other," and a general computer science (CS) faculty audience can agree that task is worth doing and doing better. An HCI researcher can say, "People can achieve this goal better and faster with my system," and the general CS faculty audience can see why that is important. When a computing education researcher says, "I now have new insights into how people understand X," the general CS faculty audience often does not know how to measure the value of that insight. "But can the students *program* using X?" and "Can they *design* systems using X?" and "X is only important in domain Y, not in the domain Z. How do you know if they can work in domain Y?"

Part of the problem is that general CS faculty do not understand education research. In social sciences, developing an insight, a hypothesis, or a "pre-theoretical" exploration is often a *lot* of work, and it is a contribution even before it is a theory, a model, or an intervention to improve some desired educational outcome.

A bigger problem is that we have many different learning outcomes and definitions in computing education. I recently told a colleague at another institution about our BS in Computational Media at the Georgia Institute of Technology, which may be the most gender-balanced ABET-accredited computing program in the U.S. (see discussion of enrollment at <http://bit.ly/1IdgDR2>, and of graduation rates at <http://bit.ly/1eNHsyE>). The response was, "That's great that you have more women in computational media, but I want them in CS."

The U.S. National Science Foundation (NSF) has been promoting two computer science courses in high schools across the country: Exploring CS (ECS, see <http://www.exploringcs.org/>) and the new Advanced Placement in Computer Science Principles (AP

CSP, see <http://apcsprinciples.org/>). ECS and AP CSP are getting significant adoption across the U.S. Because high school education is controlled by the individual states (which may delegate control down to the individual districts), not the federal government, it is difficult to make sure it is the same course in all those states. I am part of an NSF Alliance that works with states to improve their computing education (Expanding Computing Education Pathways Alliance, ECEP, <http://ecepalliance.org/>), and I am seeing states struggling with wanting to adopt ECS and AP CSP, but also wanting to fit the courses to their values and needs.

In one state we work with in ECEP, the state decided to create its own version of Exploring CS. They kept all the same technical material, but they de-emphasized inquiry learning and equitable access to computing, in order to increase the technical content. They also wanted to ensure that all high school students learn about database systems, because that was important for IT jobs in the state (they are now reconsidering this decision).

In a couple of states, there will be CS Principles and AP CS Principles as

two different courses. One reason for the difference will be the performance tasks in the AP CSP. The definition for the new Advanced Placement exam in CS Principles will include evaluation of activities in the classroom, where students describe code, create products, and demonstrate that CS is a creative and collaborative activity. It is a great idea, but implementing it takes time. The performance tasks take about 23 hours of classroom time to complete. With practice for the tasks (you would not want your students to be evaluated on the first time they tried something), AP CSP may cost a class 50 hours or more of time, which might be spent on more instructional time.

Exploring CS was developed in high school, but in some states is being adopted into middle schools. Middle schools differ even more dramatically than high schools. Some of the efforts to adopt ECS involve integrating it into multidisciplinary science and mathematics courses or units, which is common in the new Next Generation Science Standards implementations (see California examples at <http://bit.ly/1GjdgI1>). A focus on equity in computing is difficult to sustain when computing is just one of the several disciplines integrated in the course. Is the course still ECS then?

We have a babble of conflicting goals, definitions, and learning outcomes around computing education. I regularly hear confusion among the state education policymakers with whom we work in ECEP:

► “Some people tell me that there’s computer science without programming, and other people say programming is key to computer science. Which is it?”

► “We go beyond programming in our CS classes. In our first class, we teach students how companies manage their IT customer service. Isn’t that computer science? Why not?”

► “I want every student in my state to know how to write SQL for Oracle database because that’s an important job skill. Should I build that into our state’s version of ECS or our version of CSP?”

► “The workers in our state need to learn applications and tools. I see that in the job ads. I don’t see any-

“As a computing education researcher, I like the babble. I like many different possibilities being explored at once.”

body advertising for ‘computational thinking.’”

► “Can you teach computer science to our special needs students? To our English-Language Learner students? If you want to teach computer science to *everyone*, you have to cover *everyone*.”

► “Here’s our curriculum. What has to go to make room for computer science?”

Just recently, the organization that owns Advanced Placement, the College Board, struck partnerships with Code.org (<http://usat.ly/1Q4is7b>) and with Project Lead the Way (PLTW, <http://bit.ly/1M265SG>) to endorse their curricula for AP CS Principles. This is a major move to consolidate AP CSP curricula. The whole purpose for doing a new AP CS exam is to make it more accessible, to get more schools to offer it, and to reach schools and students that did not previously have access to CS education. If you were a principal of a school that never offered AP CS before and you have to pick a curriculum, wouldn’t you pick a College Board-endorsed curriculum first? PLTW is expensive for schools to offer, unlike Code.org’s curriculum, which is free. Both offer teacher professional development (PD), but Code.org pays for part of their teacher PD. It is likely Code.org’s curriculum will become the de facto standard for CS Principles.

That may be the right thing to grow computing education. Diverse perspectives are really valuable, but they are also confusing. Most school administrators do not know what CS is. The College Board is not preventing

other CSP curricula. By backing a couple of approaches, it becomes easier to figure out, “What is CS Principles? Ohhh—that’s CS Principles.” A little less babble may go a long way toward increasing access.

As a computing education researcher, I *like* the babble. I like many different possibilities being explored at once. I like a diversity of curricula, and schools with different values implementing the curricula in different ways. As a computing education advocate, I understand the education system can only withstand so much babble—especially at these early stages, when computing is still misunderstood. The teachers, principals, administrators, and policymakers who run these systems need definitions to help them understand computing. It is difficult for computer scientists to agree on these definitions. Maybe the College Board and Code.org will do it for us.

Many thanks to Barbara Ericson, Renee Fall, and Rick Adrion for comments and additions on this blog post. While our work in ECEP is funded by the U.S. NSF, the opinions expressed here are my own and may not represent NSF’s or the rest of ECEP’s.

Comments

Is there a de facto standard curriculum for AP CSA? There does not seem to be one, and that is probably good. AP CSA evolved a lot over the years while AP CSP is brand new. With code.org investing big in the largest districts, I can see the temptation to assume it will become the standard. I suspect though that schools that already have CS, especially CS without AP CSA, may want to create their own to fit their specific school environment. I think (maybe hope is more accurate) that some schools will want to experiment with different tools, languages, and curriculum in their version of AP CSP.

—Alfred Thompson

Mark Guzdial is a professor at the Georgia Institute of Technology.