Beyond Curriculum: The Exploring Computer Science *Program*

Joanna Goode, University of Oregon
Gail Chapman, UCLA
Jane Margolis, UCLA

In the past few decades, computer science has driven innovation across a variety of academic fields and become a robust part of democratic participation and the labor economy. Today's youth are surrounded with applications of these new technologies that impact how they access and produce information and communicate with friends, family, and educators. Yet, though students often gain skills as "users" of these technologies in schools, too many have been denied opportunities to study computer science and produce new knowledge required to become "creators" of computing innovations. The students who do study computer science courses often represent only a narrow band of students that excludes significant numbers of girls and students of color. Further, for a field that depends on creativity, a homogenous workforce fails to take advantage of those with diverse experiences and world viewpoints that likely foster divergent and fresh thinking.

This article will provide an overview of Exploring Computer Science (ECS), a curriculum and program developed to broaden participation in computing for high school students in the Los Angeles Unified School District. This program is framed around a three-pronged approach to reform: curricular development, teacher professional development, and policy work across a variety of educational institutions. The focus is to provide the necessary structures and support to schools and teachers that leads to high quality teaching and learning in computer science classrooms. In ECS classrooms, high quality teaching and learning is viewed within the frame of inquiry-based teaching strategies that lead to deep student content learning and engagement. The incorporation of equity-based teaching practices is an essential part of setting up the classroom culture that facilitates inquiry-based learning.

As the second largest and one of the most diverse districts in the United States, the Los Angeles Unified School District provides an important context to understand opportunities and obstacles encountered while engaging in institutional K-12 computer science education reform. This article will begin with an account of the educational research that provided key information about the obstacles students encounter in computer science classrooms. Next, we will describe the key elements of the ECS program. Finally, we will highlight several lessons that we have learned that inform the CS 10K campaign (*see* Jan Cuny's Critical Perspective "Transforming High School Computing: A Call to Action", *this issue*).

Background

The ECS program was developed in an explicit effort to address the issues of underrepresentation in secondary computer science education. The creation of this program built upon findings from a three-year, qualitative research project which examined why so few females, African Americans, and Latinos were studying computer science in urban high schools. The results, discussed more fully in *Stuck in the Shallow End* (Margolis, Estrella, Goode, Holme, & Nao, 2008), revealed a combination of structural and psychological forces that limited access, recruitment, and retention of students. Course offerings, departmental designation, teacher preparation, instructional resources, and the educational climate of school accountability required by state and federal legislation all shaped the availability and quality of computer science education opportunities for students. Interviews and longitudinal classroom observations also highlighted how educators' belief systems about who could and should learn computer science mediated student participation in computing courses. For example, often counselors and teachers served as gatekeepers to computing courses, only allowing those with a perceived "natural ability" to enroll in more challenging computing courses. This "natural ability" was often a proxy for those who had preparatory privilege at home that led to early experiences with computing. These educator belief systems can make computer science courses essentially inaccessible to many historically underrepresented students even when courses are officially "offered" at a school site.

Propelled by a common commitment to galvanize behind the findings of this study and democratize computer science education in the local region, a group of educational researchers, university computer scientists, and Los Angeles Unified district officials came together to change the landscape of computer science education in Los Angeles public high schools. Until this point, courses were haphazardly sprinkled across just a few high schools in the district, and instructors had never come together for any professional development. This interdisciplinary partnership (the Broadening Participation in Computing "Into the Loop" Alliance) collaborated to design a reform effort that would simultaneously address curriculum, professional development, policy, and equity in order to broaden participation in computing.

The first effort of our partnership was to expand on the AP Computer Science program in the district. Through district policy memos, principal outreach, ongoing teacher professional development, and the procurement of a common curriculum for all classrooms the number of classes and students dramatically increased in just two years. The number of courses increased from 11 to 23, and the number of students enrolled in Advanced Placement Computer Science increased from 225 to 611 students. Not only did the total number of students increase, but also the enrollment of girls quadrupled, Latinos quintupled, and African Americans doubled (Goode, 2007). We found that the combination of a common curriculum, intensive professional development, and district support mechanisms were effective motivation for schools to offer additional, more rigorous courses in computer science.

Yet, underlying these increased enrollment numbers was a tension that the programming-centric focus of the AP course and the advanced, college-level status was not an accessible point of entry for most students. Three years into this work, we recognized the need for a foundational high school course introducing students to the major concepts of the field of computer science, and a course that had the potential to engage the diverse populations of Los Angeles schools.

Development of ECS: A Focus on Curriculum, Instruction, & Policy

Relying on educational learning theories and research showing the inseparable link between curriculum and instruction, ECS was developed in alignment with inquiry-based design framework. Unlike the development of other computer science courses, the design of the general instructional paradigm of ECS was as important as the selection of the particular computer science content. Specifically, this course adopted an inquiry-based approach to teaching and learning computer science, an instructional strategy advocated by the National Research Council (2000).

The term inquiry is defined by the National Academy of Science in two distinct and complementary ways. First, it refers to the abilities students should develop to be able to design and conduct scientific investigations. Doing inquiry in computer science includes the ability to identify the problem and related conditions, developing some initial strategies for solving the problem, experimenting with various strategies, and examining which approach is ideal given the various contextual constraints of the problem. In computer science education, many refer to this notion of doing inquiry as computational thinking practices.

Second, inquiry also refers to a particular set of teaching and learning strategies that enable concepts to be understood through hands-on investigations. Inquiry-based teaching and learning is characterized by a student-centered approach that empowers students to help define the initial conditions of problems, utilize their prior knowledge, work collaboratively, make claims using their own words, and develop multiple representations of particular solutions. Rather than disseminating knowledge through a lecture format, teachers encourage students to arrive at knowledge claims themselves, thus owning their own learning process. Major concepts, rather than discrete collections of facts, are at the center of inquiry. This type of teaching and learning environment is particularly effective when the curriculum is project based.

It is critical to note that inquiry is not an unguided or unstructured learning approach. Rather, teachers must create structures that encourage students to ask critical questions within a particular domain of knowledge. Teachers then act as a dynamic guide, facilitating the student inquiry and asking questions of students to help steer their investigation. To be able to ask the critical questions that connect students' prior knowledge with the concept at hand, teachers must be familiar with the cultural and academic background of their students. This type of teaching, then, is a significant shift for educators who have been trained in simply disseminating content to students.

The ECS Curriculum

The curricular content of the course was decided in consultation with a group of leading computer science educators concerned with secondary computer science curriculum. Ultimately, topics were selected in an effort to make the course academically rigorous and simultaneously engaging for high school students. From the topics chosen by this group of educators, a blend of conceptually focused and experiential units were constructed to showcase both the academic underpinnings and applied practices of topics within computer science. Though the units are created in ways that allow for dynamic substitutions, the current ECS curriculum includes: Human Computer Interaction, Problem Solving, Web Design, Programming with Scratch, Data Analysis and Computing, and Robotics (Goode & Chapman, 2011). Each unit begins with the introduction of a project that culminates the learning of the particular unit.

Including culturally relevant instructional materials represented a driving focus of our course development. Following the lead of educational researchers who note that students bring rich sets of “funds of knowledge” to the classroom about homes and communities, lessons and assignments in ECS draw on students’ own prior experiences and knowledge to engage culturally diverse students in computer science. Examples of culturally situated applications of computer science are also integrated in the curricular materials to highlight the multicultural roots of computer science. These cultural design tools encourage students to artistically express computing design concepts from Latino/a, African American, or Native American history as well as cultural activities in dance, skateboarding, graffiti art, and more. These types of lessons are important for students to build personal relationships with computer science concepts and applications – an important process for discovering the relevance of computer science for their own life. Allowing students to build an authentic identity as someone who does computer science within a familiar cultural context increases the likelihood that they will pursue additional study or careers in the field (Barton and Tan, 2010).

An Intensive Teacher Support Model

Unfortunately, it is all too common for CS education reform efforts to focus narrowly on the subject matter of the discipline, with little consideration to *transforming* teacher pedagogy. Our ECS guiding philosophy is that there is a deep connection between teaching and learning, and that handing educators a new curriculum is not sufficient to support inquiry and equity-based teaching. Curriculum is more than “notes on a page” and radical transformation of teaching is required to broaden participation in computing. This requires a well-conceived system for teacher learning.

For these reasons, the ECS program closely couples the curriculum with on-going teacher professional development and community building. The aim of this program is to help support teachers’ learning of the ECS content while simultaneously strengthening teachers’ pedagogical content knowledge that supports the teaching of ECS. For many teachers, building a pedagogy that supports inquiry-based teaching and equity practices in the classroom is a multi-year process. This ongoing learning process with colleagues and university educators leads to a professional teaching community - another key aim of the ECS teacher support model.

The ECS professional development program begins with a weeklong summer institute where teachers work through the first half of the curriculum, enough content for the first semester of the course. During the academic year, quarterly follow-up Saturday sessions are held to focus on the remaining three units of the curriculum. Thus, the Summer Institute and quarterly follow-up sessions are not separate events, but part of a rolling system of whole-group professional learning workshops. During these sessions, teachers alternate planning and team-teaching particular lessons as ways of learning content, practicing inquiry-based teaching methodologies, and having particular teaching moments serve as fodder for debriefing discussions about strategies in teaching computer science within diverse learning contexts. This ongoing model of professional development also provides teachers an opportunity to meet on a regular basis and share their classroom teaching experiences while simultaneously learning new curricular content and pedagogical approaches.

Providing ongoing professional development for this community of teachers has highlighted the importance of continuing support past the first year teachers participate in the program. Once they have taught all the way through the curriculum the first year, teachers are significantly less concerned about learning the content of the course and are able to devote more attention towards changes in pedagogy that improve teaching and learning in the classroom. Furthermore, they are in a professional space that is rich for reflection on teaching successes and challenges in their own ECS classroom. Teacher education literature notes that this type of professional reflection is key to growth in professional practice (Howard, 2003). The importance of this two-year model of professional development for ECS cannot be overemphasized.

Though our research has shown that the ECS summer institutes and weekend workshops are very helpful in supporting ECS teachers, the data also concurs with findings that reveal how summer professional development workshops alone are insufficient to transform classroom practice, and how in-classroom follow-up learning is necessary. As a result we established an in-classroom coaching program and teacher inquiry groups. Coaches and researchers conduct regular visits in the ECS classrooms to model teaching, engage in co-teaching planning discussions, and hold reflection conversations with teachers. Along with the professional development workshops, these classroom visits provide the types of interaction teachers might have if they were not the sole computer science teacher at their school.

Policy

A critical part of the ECS reform initiative is tackling the institutional issue of placement of computer science within the traditional curricular canon. This is not a challenge to be taken lightly, considering that the core academic subjects defined by *No Child Left Behind* are the same curricular domains defined a century ago during the industrial age (Tyack & Cuban, 1995). Updating the "core curriculum" to reflect 21st century knowledge, requires multiple levels of support – from state officials, administrators, counselors, and teachers. Without a clear "fit" into the common core, computer science classes are often relegated to low-level training domains and lose much of the academic content. Relatedly, few states offer computer science teaching credentials. Without a credentialing program, there are limited ways to learn the teaching methodologies effective for instructing computer science. Thus, as part of any reform effort, local and state policies need to be addressed to work towards sustainability in the schools.

On the local level, the ECS program relied on our partnership with the school district to learn the local communication and decision-making process. In LAUSD, much of communication came in the form of memos for school board members, local area superintendents, and principals. We used this system to distribute information about the program and to recruit interested schools. One of our team collaborators serves on a principals leadership group, and he uses his knowledge of local schools and personal relationships with other administrators to place ECS classes in school's course offerings schedule. Once a teacher is identified at that school, they become part of the professional development program and are given additional tools to help with recruitment at their school site.

At the state level, we acknowledged the need to give academic credibility to ECS. To this end, we engaged in an approval process with the University of California Office of the President, requesting status for ECS as an officially recognized elective with "college preparatory" status. This status was awarded, and though we initially applied to be an elective in the mathematics category, the ECS class was assigned elective status in the Career Technical Education (CTE) category. Though functionally this assignment is not important for students, it signified an important opening for ECS to fit into multiple pathways in the high school curriculum. In a sense, this is a course that counts in two different, complementary ways – as a college elective and as a career-technical education course. Rarely does a single course satisfy both of these criteria and attract two different school populations to the same academic class. Positioning ECS as a CTE course will help strengthen ECS sustainability. CTE is becoming more central to California secondary education, and the California CTE leadership has also been a strong proponent of ECS. CTE is interested in using ECS to anchor a sequence of CS courses that can be brought into the schools. Additionally, CTE is a good source of potential funding for resources, such as robots, and capacity building such as PD expenses. This is critically important in a time of financial crisis of most urban school districts.

Impact of ECS

Measuring success in equity-based computer science education reform can be examined through student access to the course, changes in student engagement and interest with computer science, and the amount of computer science content learned as a result of a particular class or program of study. Currently, we are investing much of our project resources into designing measures to assess student learning of both CS concepts and the problem-solving at the heart of ECS. However, we have used student and teacher surveys, interviews, district enrollment data, and field observations to examine issues of access, engagement, and interest in ECS.

In the first four years the course was offered in Los Angeles schools, enrollment has increased significantly. Following a pilot in five schools involving 306 students, enrollment tripled to 922 students in 2009-10. In the past three years, course enrollments have continued to increase at a rate of 50% per year, and now, over 2000 students are enrolled in ECS in the 2011-12 school year. But, our goals of broadening participation go beyond looking at increases in student participation rate and require an examination of participation rates of traditionally underrepresented students. Importantly, girls represent 40% of ECS students, African Americans represent 10% of enrollment, and Latinos represent 71% of enrollment. In LAUSD, Latinos are 73%, African-Americans 10%, Asians 6%, and Whites 8% (www.ed-data.k12.ca.us).

An analysis of student responses about their interests and engagement in the beginning and the end of the course show students significantly increased in their motivation to pursue and engage in computer science and related activities over the course of the year, as measured by the *Computer Science Attitude Survey.* Selected findings include:

- The number of students reporting being "Very interested" in learning more about computer science increased from 17% to 43% after taking ECS.
- Over half of the students (60%) responded that they could envision themselves working in a technology-related field in the future. In addition, 44% of participating students reported being interested in a computer-science related field as a future career.

Asking students to describe what they learned in their own words cast a more descriptive light on student experiences in ECS. At the end of the year survey, students were asked to complete the phrase, "Because of ECS, I….". A sampling of answers which highlight engagement in computing concepts, inquiry approaches, and creativity include:

- *[Am] more in tune with how everyday things that effect might life are made and programmed*
- *Learning how to be specific*
- *Capable of learning and adapting to problems and learning its solution*
- *More imaginative*
- *Quite a nerd and happy I got this class*
- *I now have patience. I'm not scared or shy to speak my mind or ask questions*
- *Thinking more logically and always trying to get around and overcome obstacles in my way*
- *Because of computer science I am now more experienced at something I didn't even know existed*
- *More aware of the computer science world. I now know what it takes to be part of it and how difficult yet fun it can be*
- *Able to recognize I am a problem solver and problems can be solved in a variety of ways*
- *Able to think outside of the box. Instead of just thinking everything just happens I know there is a lot of hard work that is put into everything that happens.*
- *I'm more creative and I don't give up when something that I don't completely understand is difficult. I keep trying and see the results.*
- *Smart ^.^*

Lessons Learned From ECS

The systemic approach ECS has taken to reform computer science education in Los Angeles has been met with multiple successes, some setbacks that result from the larger educational crisis, and lessons learned. We have documented this process in an effort to provide our process as a case study for other local areas wanting to reform computer science education in culturally diverse schools, while recognizing that each local situation differs in important ways. Yet, there are many commonalities as well. As the computer science community considers the CS 10K Teachers Campaign, the ECS project offers suggestions based on lessons learned in the Los Angeles reform program.

1. *Investing in Local Partnerships*

Navigating the bureaucracy and structures of local education agencies takes both time and a "fit" within the many mandates and needs of local schools. Aligning with existing reform programs can be a good entry into schools. Depending on the particular local context, effective communication mechanisms must be designed to facilitate true cooperative decision-making on key reform elements by district officials, educational researchers, and computer science educators. Building a network of teachers, administrators, and school officials who are invested in the program can address issues of institutionalism, and ultimately sustainability, within the school system. This approach requires a "bottom-up" approach to impacting policy and requires relationship building on multiple fronts and settings with a variety of educators and policymakers.

Recruiting local schools and teachers to participate in computer science education programs is made easier when professional and material resources needed to launch a successful and academically rigorous computer science class are made available at no initial expense to schools. For administrators, the lure of a university-developed common district curriculum, extensive teacher content and pedagogy support, and class sets of computing resources provides a cohesive framework of support that makes the initial offering of the class less risky. With a successful launch, ongoing institutionalization to sustain the course is more likely to take place.

2. *Talking about "Quality Teachers" – The Elephant in the Room*

The reform effort in Los Angeles highlights the importance of providing teachers ongoing professional support to reflect upon and improve their instructional classroom practices. Classroom observations highlight the vast differences in teaching styles and teaching effectiveness in the classroom. Our rich ethnographic data, that we have been gathering through classroom observations, surveys, and interviews, beautifully illustrates that simply placing a teacher in a computer science classroom – even with supportive professional development – does not ensure the type of high-quality teaching that excites and motivates youth. Computer science education reform efforts must seriously address issues of teacher quality and teacher effectiveness – particularly for working with culturally diverse students. Otherwise, we run the risk of dull pedagogy or culturally insensitive practices continuing to turn students away from the computing discipline.

Certainly, there is extensive evidence in educational research literature showing that quality teaching is the most significant variable for determining student achievement and thus have a critical effect on determining life opportunities (Darling-Hammond, 1997; 2002). We cannot confuse mastery of computer science with competence in classroom teaching. In fact, we have found that few of our ECS teachers had already mastered CS content when entering the program. And, some of our strongest teachers are from other non-STEM disciplines such as social studies and English. While having a strong computing background is an absolute plus, it is clear that the CS10K efforts should not be limited to those who have already mastered the content.

3. Appropriate Measures of Reform Success Must be Used

Disrupting the traditional secondary curriculum with the introduction of computer science education can entrap reformers into proving that other subjects benefit as a result of the knowledge transferred from this class (i.e. mathematics or science achievement) or that students are more likely to choose to enroll in computer science classes in the future. These are incredibly high, if not impossible standards for any one high school class to obtain, and the high possibility of confounding variables makes it difficult to ultimately make any conclusions. Additionally, these types of measures dilute the importance of computer science as essential knowledge for 21st century students in its own right. For instance, geography educators are not required to prove that geography increases test scores in other subjects, nor are they asked to assure students go on to major in geography in higher education. Instead, gaining the knowledge of geography itself is seen as an important goal of contemporary schooling as preparation for being a responsible global citizen. As a community of computer science educators, we must allow access to learning computer science education in high school serve as a fundamental right to learn, rather than a stepping-stone for a future purpose.

4. Equity Must Go Beyond Access

While increasing the number of courses and teachers is a critical step, equity will require more than access. Computer science is a field that long been identified with a narrow band of white and Asian males---especially those who have had the resources to start their exploration with computing at a young age. The learning environment of the more advanced computer science classrooms has supported the culture of these students and often made others to feel as "outsiders," as if their concerns, perspectives, were not valued in the field. Having assignments, projects, and dialogues that value the concerns, identities, and viewpoints of a diverse student body is important to building real equity in a classroom.

Addressing equity at the classroom level, then, requires an examination of equity-based teaching practices. Such practices validate all voices and perspectives in the classroom, particularly those that reflect culturally diverse experiences. These practices include a vision of success for all students, and instructional strategies that foster the participation of all learners in the classroom setting. This attention to equity at the classroom level, and not just at the "access" level, is essential for authentically broadening the participation in computing.

Conclusion

Broadening participation in computing at the local level requires a sustained, long-term effort that builds and leverages partnerships to expand opportunities for students to learn computer science. But, as we have found, access to courses is not enough. Rather, our model of broadening participation in computing focuses on increasing access to high quality and culturally-relevant curriculum and teaching in classroom settings populated by

culturally diverse students. Simultaneous policy work ensures the academic credibility and placement of this class within rigid course offerings of high schools.
As we move forward, our ECS program has begun to focus on two of the outstanding and challenging issues facing the CS10K movement:

1) Quality teaching---What does it look like? How can it best be measured?
2) Assessment of student learning---What should be measured and how best to do it?

These issues combined with what we know about curriculum development, teacher professional development and community building, CS pedagogy, inquiry and equity instructional practices, and policy changes, all are critical components to building an effective program for broadening participation in computing. In the call to expanding computer science education, ECS serves as one example for working authentically in partnerships with public schools at the ground level in culturally diverse school classrooms.

Acknowledgements
This project is supported by the "Broadening Participation in Computing" program of the National Science Foundation.

References

Barton, A. and Tan, E. (2010). We Be Burning'! Agency, Identity, and Science Learning. *The Journal of the Learning Sciences*, *19*(2): 187-229.

Darling-Hammond, L. (1997). Doing what matters most: Investing in quality teaching. NY: National Commission on Teaching and America's Future.

Darling-Hammond, L. (2000). Teacher quality and student achievement: A review of state policy evidence. *Education Policy Analysis Archives, 8*(1).

Goode, J. (2007). If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. *Journal of Educational Computing Research, 36*(1), 65-88.

Goode, J. & Chapman, G. (2011). *Exploring Computer Science, 4TH ed*. Los Angeles, CA: Computer Science Equity Alliance.

Howard, T. C. (2003). Culturally relevant pedagogy: Ingredients for critical teacher reflection. *Theory into practice, 42*(3), 195.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.

Moll, L., Amanti, C., Neff, D., & Gonzalez, N. (1992). Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory into practice, 31*(2), 132-141.

National Research Council. (2000). *Inquiry and the National Science Education Standards: A Guide for Teaching and Learning.* Washington, D.C.: The National Academies Press.

Tyack, D., & Cuban, L. (1995). *Tinkering toward utopia: A century of public school reform*. Cambridge, MA: Harvard University Press.

Author Information

Joanna Goode
Department of Education Studies
College of Education
5277 University of Oregon
Eugene, OR 97403-5277
goodej@uoregon.edu

Gail Chapman
UCLA, Leadership and Professional Development
Exploring Computer Science
Moore Hall 1320
405 Hilgard Avenue
Los Angeles, CA 90095-1521
chapgail@gmail.com

Jane Margolis
UCLA, Graduate School of Education and Information Studies
Exploring Computer Science
Moore Hall 1041E
405 Hilgard Avenue
Los Angeles, CA 90095-1521
margolis@gseis.ucla.edu

Categories and Subject Descriptors: K.3.2 [Computers and Education]: Computer and Information Science Education – Computer science education, Curriculum

General Terms: Experimentation, Human Factors, Design, Management, Measurement

Keywords: Computer science education, pedagogy, Exploring Computer Science