Exploring Computer Science: A Case Study of School Reform

JOANNA GOODE, University of Oregon
JANE MARGOLIS, University of California, Los Angeles

This article will detail efforts to broaden participation in computing in urban schools through a comprehensive reform effort of curricular development, teacher professional development, and policy changes. Beginning with an account of the curricular development of *Exploring Computer Science*, we will describe the inquiry-based research that underlies these learning materials. Next, we argue that accompanying professional development that supports the curriculum is essential for supporting this inquiry-based approach to computer science instruction. We then explain the policy strategies used to designate this course as a college-preparatory elective and place it in 17 Los Angeles high schools. Finally, we share the initial results of how students experience this course and ongoing challenges encountered when working in the public school system. The article concludes with a discussion of how longitudinal reform effort requires a strong foundation and deep roots to successfully democratize computer science education.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education; curriculum*

General Terms: Human Factors

Additional Key Words and Phrases: Gender and ethnicity, exploring computer science, broadening the participation in computing, pedagogy, curriculum

ACM Reference Format:
Goode, J. and Margolis, J. 2011. Exploring computer science: A case study of school reform. ACM Trans. Comput. Educ. 11, 2, Article 12 (July 2011), 16 pages.
DOI = 10.1145/1993069.1993076 http://doi.acm.org/10.1145/1993069.1993076

1. INTRODUCTION

Historically, a very narrow band of students—predominately, but not exclusively, from families able to provide computers, Internet access, robotic kits, plethora of software, and parental knowledge—have been the ones who have had an early jump start and engagement with computer science, leading to majoring in the field and/or entering computer science careers. In this article we discuss how our NSF Broadening Participation in Computing (BPC) project, Exploring Computer Science, has attempted to "democratize computer science" to make computer science knowledge more available to and engaging for a broader segment of our student population.

Our strategy has been to bring computer science learning into the precollege curriculum, especially into high schools with high numbers of traditionally underrepresented students of color. To do so we have had to address an array of issues, including

The authors acknowledge the National Science Foundation's "Broadening Participation in Computing" program for providing the funding to support this work.
Authors' addresses: J. Goode (corresponding author), Department of Education Studies, University of Oregon, 125D Lokey Education Bldg, 5277 University of Oregon, OR 97403-5277; email: goodej@uoregon,edu; J. Margolis, Department of Education, University of California, Los Angeles, CA.

DOI 10.1145/1993069.1993076 http://doi.acm.org/10.1145/1993069.1993076

the absence of engaging high school curriculum and pedagogy, inadequate teacher professional development, toxic stereotypes and belief systems within the schools about which students can (and cannot) do computer science, the absence of a computer science certification program for teachers, and why computer science exists on the margins of many public school core requirements. Further, we have had to respond to the national educational crisis, as computer science does not exist as an island unto itself. Any attempt to broaden participation in computing must address deep systematic educational inequity that runs throughout our entire educational system. And, despite these issues, we believe that the struggle to get engaging computer science into the schools—because that is where the students are—is a critical step to expanding access to computer science knowledge and learning opportunities.

This article describes the details of how a BPC K-12/University partnership has carried out a mission to increase access to quality computer science education opportunities in the second largest and one of the most diverse school districts in the United States, the Los Angeles Unified School District. We will discuss:

— The content, learning theory, and philosophy behind the high school curriculum we developed, *Exploring Computer Science* (ECS);
— The ECS teacher professional development model, including the coaching program and building of teacher community;
— Our assessment of the first ECS pilot year and its impact on students and teachers;
— The necessary strategy and policy changes to sustain these efforts locally and nationwide.

2. THE STORY OF EXPLORING COMPUTER SCIENCE

The Computer Science Equity Alliance (CSEA)—the alliance we formed that includes educational researchers, university computer scientists, K-12 school district officials, and leadership from the Computer Science Teachers Association (CSTA)—began focusing on equity issues in computer science education in Los Angeles Unified School District (LAUSD) in 2004. The CSEA was formed in response to a three-year qualitative research study that highlighted instructional and structural barriers to computer science education especially for females and students of color [Margolis et al. 2008; Goode et al. 2006]. Originally focused on increasing access to Advanced Placement Computer Science (APCS), especially in schools with high numbers of students of color, the Alliance successfully doubled the number of APCS courses in the LAUSD within two years. Each summer from 2004–2007, the Alliance offered a weeklong professional development in the form of a Summer Institute for Advanced Placement Computer Science teachers. An AP Readiness Program was launched so teachers can bring their students to the UCLA campus once a month for supplemental instruction and embedded professional development. These monthly meetings became a place for APCS teachers to continue to collaborate and build a teaching community. This approach of increasing classes and providing instructional and pedagogical resources and structural support to teachers led to staggering results—the number of students studying APCS in LAUSD tripled—girls quadrupled, Latinos quintupled, African Americans doubled [Goode 2007].

Despite these results, our formative and summative research identified a significant tension between the Alliance's goals of increased access and engagement with computer science for a broader segment of the student population and the goals of the APCS course. Teacher interviews revealed that as a first course in computer science, the learning curve was too steep for the students to learn APCS content in time for the May exam. Teachers also reported that the course's exclusive focus on program methodologies in Java failed to attract the interest of students who were not already enamored with computing. It was clear from conversations with district officials and school administrators that the APCS enrollment increases were successful largely due to the college-preparatory status of the course for college-bound students. This status is an important indicator of academic rigor, but also, students' course schedules are so impacted by graduation and college-preparatory requirements that they are often not able to take any noncollege-preparatory electives. It seemed that the success of the computing reform efforts were due to the academic status of the APCS course, not the particular content. At this point, university computer scientists joined this conversation and offered a host of computing topics that were more likely to tap into the interests of students. It became clear that in order to reform computing in Los Angeles, a new college-preparatory course must be offered that introduces students to exciting introductory topics in computer science within an instructional framework informed by educational learning theories focused on increased access and equity.

2.1 Exploring Computer Science Curriculum

In 2008, recognizing that no college preparatory computer science course besides APCS existed, we took the bull by the horns and decided to design a course which could precede an APCS course. Designing curricular materials for a new foundational content-area course requires deep knowledge in computer science content, pedagogy, learning theory, and the communities from which students draw their informal knowledge about the world [Bransford et al. 1999]. The resulting curriculum, *Exploring Computer Science,* reflects a selection of computing topics that are accessible and related to the academic interests of urban high school students. The instructional framework of the ECS course was developed as a collaborative effort between university education researchers and LAUSD. Based on the ACM model course guidelines, and drawing from specific learning theories that support inquiry-based learning, the curricular materials require students to engage in computational thinking and complex computing projects and it weaves assignments that pull from diverse communities of interests throughout the curriculum. Ongoing opportunities for collaboration, communication, and multiple ways of knowing are embedded into the design of the course. It is important to note that the instructional framework presented in this curriculum is closely aligned with similar science education reform efforts in LAUSD, an effort well known by principals and other key curriculum decision makers.

The list of ECS course topics was initially framed in an informal conversation at SIGCSE 2007 with leading K-12 and university computer science curriculum experts. With a general agreement of course topics and the sequence of these topics, educational researchers shared this list with LAUSD mathematics and science leaders and UCLA computer scientists who work closely with local high school computer science teachers and students. The scope and sequence outline was further revised after this additional input.

The resulting course is composed of six instructional units: (1) Human-Computer Interaction; (2) Problem Solving; (3) Web Design; (4) Introduction to Programming; (5) Robotics; (6) Computing Applications. Each unit averages six weeks in length. The course reflects topics and objectives drawn from Levels II and III of the ACM's K-12 curricular framework [Tucker et al. 2003]. Each unit begins with a discussion of a final unit project so students have an appreciation of the problem they are working towards solving while learning the necessary skills required to solve the problem. Assignments and instructional strategies weave through the six units, conceptually linking the topics together. The curricular materials include original assignments as well as lessons drawn from other computing education materials, including Scratch [Maloney et al.

2008] assignments, Lego Mindstorms robotics challenges, and Computer Science Unplugged [Bell et al. 2009] activities.

Though the ECS curriculum includes a structured list of scope and sequence of foundational computing topics, the materials were developed around an effort to engage students in practices that build computational thinking skills and understandings. Rather than simply being a list of discrete knowledge, the overarching goal of this course is to help students gain knowledge about the capabilities of humans working with computers to analyze and solve problems and conduct a variety of tasks.

The organization of the ECS curricular materials makes each unit amenable to dynamic substitutions or modifications as engaging new computing topics are brought into the curriculum. Importantly, this dynamic design is necessary for students and teachers to keep up with the rapid pace of computing innovations. For example, a Computing Applications unit currently draws from seismic data provided by UCLA's Center for Embedded Network Sensing (CENS). However, more recent CENS technology is focusing on how students can use smartphones to gather and analyze data about their community. The development of this updated unit and accompanying technological platform is currently in progress and will be fully integrated into the curriculum in summer, 2011.

In 2008, five teachers, including two LAUSD teachers, contributed sample lessons for particular curricular units in the course. Each of these teachers was provided the scope and sequence of the entire course, the instructional framework, and a list of objectives for their particular unit. The teacher writers drew from their own knowledge and teaching experiences to contribute towards a particular computing topic. The authors of the curriculum integrated these ideas into a cohesive, six-unit curriculum designed to take place over a two-semester school year. Daily teacher instructional plans and supplementary resources are included in this curriculum.

2.2 Focus on Inquiry-Based Instruction

The *Exploring Computer Science* [Goode and Chapman 2009] curriculum focuses not only on computational topics, but also draws from leading educational theories about how students learn. Often, the attention given to reform efforts focuses on the curricular or policy levels and fails to address the impact on actual teaching practices in the classroom. Given the rich cultural diversity of students and the importance of classroom teaching on student achievement, we must go beyond curricular efforts (with accompanying professional developments) and focus on how reforms can improve *classroom teaching*. Educational research points to the power of inquiry-based instruction in providing students rich opportunities to learn in science domains. For example, research studies in urban and suburban areas have shown that middle school students who learned physics in an inquiry-based setting outperformed their high school counterparts who were taught with more conventional methods [White and Fredrickson 1997; White and Frederiksen 1998].

Inquiry-based instruction is understood to be an educational method that encompasses a range of theoretical, curricular, and pedagogical approaches to learning. Inquiry-based instruction relies on: (1) a curriculum based on core concepts that students understand through induction rather than memorization, (2) teaching based on guided inquiry rather than didactic instruction, and (3) assessment that is open-ended rather than standardized. The roots of inquiry-based instruction can be traced to Dewey, who argues that problems students solve in schools should be spurred by children's natural curiosity about the world around them [Dewey 1938]. Vygotsky deepened these ideas by proposing a zone of proximal development, which represents "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance, or in collaboration with more capable peers" [Vygotsky 1978; p. 86]. Vygotsky argued that the role of education is to provide rich learning experiences within children's zones of proximal development. That is, schools should scaffold students understandings based on experiences. Inquiry-based instruction strives to meet this challenge.

Inquiry has been central to STEM-based reform efforts for two decades. Learning scientists often focus on scientific domains for models of teaching and learning that raise student engagement and achievement. Research on the effectiveness of inquiry-based learning has emerged from this rich domain of research on student learning [Bransford et al. 1999]. The policies and standards in science education, in particular, are critical in contextualizing this effort for our similar, but more local, efforts in computer science education.

2.3 Focus on Culturally Relevant and Meaningful Curriculum

The ECS efforts in Los Angeles are driven by a commitment to create learning resources that draw from the rich experiences that a diverse student body brings into the classroom. Far from blank slates, our research has demonstrated how students arrive in computing classrooms with rich community knowledge and informal computational thinking approaches [Margolis et al. 2008]. Certainly, the focus on community in ECS is apparent throughout the instructional units, often as part of an applied project. In one unit, students build Web sites about community issues that concern them. In a final unit project in programming, students use Scratch to develop stories about their community by highlighting positive attributes, recommending something that should change, and incorporating statistics to support their narrative.

A new instructional unit, piloted with six teachers in spring 2010, will engage students in a data analysis unit utilizing a mobile community-based investigation. We will provide smartphones to ECS students to collect data on their community. Developed in collaboration of the Computer Science Equity Alliance and the Center for Embedded Network Sensing, this unit will support students' engagement in a project focused on the assets and stressors in their local community.

The ECS equity model borrows from research in other science and mathematics education efforts that have developed culturally relevant instructional activities. For instance, Louis Moll's work studies how curricular reform can result in more culturally relevant curriculum for students to learn science. His work focuses on the "funds of knowledge," or the cultural and community knowledge that students bring with them into the classroom. Working with teachers, Moll et al. [1992] demonstrate how blending traditional science curricula with cultural influences students receive at home leads to increased learning for Latino students. Though not dealing with structural issues of science education, this type of curricular reform is essential for improving STEM education for underrepresented students.

Another example of a reform effort centered on STEM equity issues can be found in the Algebra Project. Moses and Cobb [2001] describe how the national Algebra Project exemplifies the advantages of linking culture to learning materials and pedagogy for African American and Latino mathematics students. The mission of the Algebra Project is U.S. mathematics literacy effort aimed at helping low-income students and students of color successfully achieve mathematical skills necessary to ensure "full citizenship in today's technological society." As part of their effort to reform mathematics education, the Algebra Project has developed curricular materials, trained teachers and teacher educators, and provided ongoing professional development support and community involvement activities to schools seeking to achieve a systemic change in mathematics education. The project follows a three-pronged framework to their reform. First, they rely on the research and development of leading mathematicians to ensure that students and teachers are learning content and instruction based on research findings. Second, the Algebra Project focuses on building the professional capacity of mathematics teachers at school sites, offering ongoing professional development for educators. Third, the project relies on community and site development that models the efforts employed during the civil rights movement. The Algebra Project highlights the importance of working with teachers and developing community connections to address belief systems about studying mathematics. Taken together, this reform effort has reached approximately 10,000 students and approximately 300 teachers per year in 28 local sites across 10 states.

Additionally, we have integrated the culturally situated design tool work of Ron Eglash into the ECS curricular resources [Eglash et al. 2006]. LAUSD students are 75% Latino/a, 11% African American, 8% White, 4% Asian, 0.3% American Indian. We believe that contextualizing some of the fundamental computational thinking principals within an exploration of cultural roots and artifacts will be very engaging for the LAUSD student body.

3. THE STORY OF ECS TEACHER PROFESSIONAL DEVELOPMENT

3.1 Summer Institutes and Quarterly Workshops

One of the most important things we have learned in this work is that the teacher professional development is an essential companion to the ECS curriculum. For our efforts to broaden participation in computing, teacher professional development sessions must address pedagogy, belief systems that perpetuate deficit thinking and stereotypes about which students will excel (and which will not) in computer science, in addition to content. In order to offer the course at a school site, principals must send a teacher to the ECS Summer Institute and weekend workshops. Participating teachers receive a stipend of $200 per day for their participation in whole-group professional development events.

In 2008, six teachers attended the ECS Summer Institute in preparation for teaching the pilot course in 2008–09. In 2009, 23 teachers from 20 schools participated in the Summer Institute; 17 of these teachers ended up teaching an ECS course the following year. These culturally diverse teachers represent the geographical diversity of the Los Angeles basin. Teacher reflections from the ECS Summer Institute radiate a sense of passion and justice for providing accessible computer science education to students of color attending large, public schools. Teachers report a shared sense of community amongst one another and an awareness of explicit pedagogical approaches that create places of academic hospitality.

Given the instructional design of the curriculum, it is important to focus on and explicitly model the inquiry-based design of the course in teacher workshops. To this end, several instructors take turns facilitating the professional development activities and publicly reflecting on their own teaching. As a result, teacher participants report that they had developed a deepened understanding of how computer science content can be taught through role playing, jig sawing activities, pair and small-group collaboration, structured tinkering, multiple solutions, utilizing manipulatives, simulations, English language learner modifications, proactive recruitment of females, journal reflections, and interdisciplinary connections. Rather than providing the answers, the professional development instructors facilitate participant discussions of problems and draw from the rich and varied expertise in the classroom.

A unique feature of the professional development is the distributed nature of expertise about teaching, computing knowledge, and the culture of Los Angeles high school students. No one participant or instructor in the professional developments has the singular solution for broadening participation in computing; rather, the professional development becomes a space where conversations about teaching wisdom and innovative computing topics intersect. Teachers, some who have been teaching APCS for years, join other teachers who have powerful pedagogical ideas but need to build their computing content knowledge. University computer science educators and veteran practitioners facilitate the conversation and continue to learn from the classroom teachers who connect with students daily. Important professional relationships have been built amongst colearners and a computing community of practice has been cultivated and continues to grow with each subsequent workshop. Evaluations of teacher participants point to the desire to bring this academic accessibility and hospitality to their own classroom community. One teacher noted with appreciation that the summer institute was like a "think tank."

Since the ECS Summer Institute is only one week, participants work through the first three and a half units of the curriculum, enough content for the first semester of the course. During the academic year, quarterly follow-up Saturday sessions are held during the school year 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 sessions. This ongoing model of professional development also provides teachers an opportunity to meet on a regular basis and reflect on their classroom teaching experiences while learning new curricular material.

The close link of the curriculum to professional development is a necessary coupling for introducing a new course in high schools. In this model of inquiry-based reform, the effectiveness of the course is limited if not accompanied by high-quality professional development that focuses on content and instructional strategies. Though research has shown that the ECS Summer Institutes and weekend workshops are helpful in supporting ECS teachers, interviews and observations reveal how traditional professional development institutes and workshops alone are insufficient to transform classroom practices. Off-site professional development events are simply not sufficient supporting teachers in crafting their pedagogical approaches around a new curriculum. As a result we have established several other opportunities for teacher learning including a coaching model and teacher inquiry groups. Along with the whole-group professional developments, these professional learning activities are meant to provide the types of informal interaction teachers might have if they were not the sole computer science teacher at their school.

3.2 Coaching for Inquiry-Based Teaching

A new computer science coaching program, provided through additional NSF support, further supports ECS teachers in their classrooms by providing content tutorials, coteaching opportunities, model teaching lessons, formal observations, collaborative planning sessions, and instructional support as needed. Two ECS coaches serve as resources for teachers in setting instructional goals, having opportunities to work towards those goals, and receiving feedback. Within this process, the coaches encourage teachers to continually reflect and document their own learning processes.

Our model of instructional coaching borrows from the work of Knight, who considers instructional coaches as on-site professional developers who work collaboratively with teachers to help them incorporate research-based instructional practices in the classroom [Knight 2007]. Knight's model of instructional coaching includes a cycle of seven practices: enrolling the teacher into a coaching program through a one-on-one interview, engaging in collaborative planning and goal-making, modeling a lesson by the coach, debriefing the lesson, observing the teacher teach a lesson using a different instructional strategy, and debriefing the teacher's lesson [Knight 2008]. The observations are done using a coconstructed observation form. According to Knight, this form of coaching is grounded in the seven principals of equality, choice, voice, dialogues, reflection, praxis, and reciprocity.

To date, eight of the seventeen ECS teachers have agreed to participate in a more structured instructional coaching program during the second semester of Exploring Computing Science. In this model, each teacher will select three coaching activities to participate in over the course of the semester. These activities include planning conversations, focused observation and feedback, coteaching, videotaping and reflection, visiting a colleague's classroom, and analyzing student work as evidence. Before and after each activity, the teacher will debrief with the coach and discuss their expectations, learning experiences, and progress towards their self-selected instructional goals.

The research on instructional coaching shows great promise at bolstering the results teachers might gain from attending professional development workshops alone [Knight 2008]. A study of 50 teachers examined the differences between teaching practices for teachers who attended an after-school workshop and teachers who attended the workshop and provided follow-up instructional coaching. The results showed a significant increase in teacher adoption of instructional strategies presented at the workshop for teachers who engaged in the instructional coaching program. Rather than replacing more traditional workshop-based professional developments, instructional coaching is viewed as a way of helping teachers connect the theory to practice in their classrooms. Certainly, this strategy holds great promise in helping teachers shift towards a more inquiry-based form of teaching.

We have also begun to organize and support smaller ECS teacher inquiry groups that meet outside of school, for now based on geographical proximity. In these meetings teachers can share teaching strategies and/or discuss problems they are encountering, and solutions they have devised. To assist teacher learning community building efforts we have set up a Ning social networking site.

4. INITIAL RESULTS

4.1 Initial Pilot Results

A total of seven ECS teachers initially piloted the course during 2008–2009. Almost 300 students enrolled in the pilot courses across these seven comprehensive high schools. Females accounted for 42% of course enrollment and African American and Latino students made up 92% of course enrollment. Semistructured teacher interviews, teacher surveys, student surveys, and observations informed specific and general revisions that were needed in the curriculum and a major overhaul took place in 2009. The pilot study revealed promising findings about the effectiveness of ECS for students. Using precourse and postcourse survey instruments, research showed that participation in the course resulted in:

- Increasing students' perceptions of the usefulness of computer science ($p<.05$);
- Increasing students' beliefs about the appeal of computer science ($p<.01$);
- Increasing students' perceptions of computer science as enjoyable and stimulating ($p<01$);
- Increasing students' motivation to stick with a difficult problem ($p<.001$);
- Students' willingness to have the answer given to them instead of working it out decreased ($p<.05$);

Table I. Exploring Computer Science Enrollment, 2009–2010

Race/Ethnicity	Females	Males	TOTAL
Latino	18	25	43
African American	4	7	11
Asian	28	54	82
White	18	25	43
Filipino	4	7	11
Pacific Islander	0	2	2
Unknown	-	-	45
TOTAL	342	534	921

Further, students reported that they were more likely to take computer science courses after high school and pursue computer science as an academic major after enrolling in the course.

4.2 Scaling Up

In the 2009–10 school year, 17 schools offered the ECS course to 921 students. Most of these schools enroll a majority of Latino and African American students. Enrollment statistics featured in Table I demonstrate the rich diversity of students enrolling in this class. Importantly, Latinos account for 70% of student enrollment and African American students account for 10% of student enrollment. Girls made up 37% of the student enrollment.

Preliminary analysis of 2009–2010 evaluation data illustrates the effectiveness of the course for the larger group of students. Precourse and postcourse surveys were drawn from 672 and 315 students, respectively. The smaller postcourse numbers are a reflection of the difficulty of collecting survey data from classrooms busy wrapping up the school year. The survey was expanded from the pilot year to include questions on student knowledge in addition to questions on interest and engagement.

Two items concerning interest and engagement were found to be significant when comparing these two surveys.

— Increased students' perseverance for working through a computer science problem or assignment ($p < .05$);
— Increased students agreement that computer science is enjoyable and appealing ($p < .05$).

However, we also found that many other interest and engagement variables listed in the pilot results described before, demonstrated no measurable differences during the time of ECS enrollment. Further analysis of this data, however, shows that many students who enrolled in ECS initially expressed a high interest in computer science before taking this course, making it difficult to report a statistically significant increase across large numbers of students by the end of the course.

Importantly, students reported an increased understanding of computing topics from the beginning of the course to the end of the course. Specifically, students scored their knowledge of particular topics significantly higher in all six areas presented on the survey: familiarity with the different fields computers are used, problem-solving techniques, creating Web sites, programming a computer, robotics, and computer modeling. These responses were significant at the $p < .01$ level.

Taken together, these promising results suggest how ECS is shaping students' understandings and appreciation of this discipline.

5. SUSTAINABILITY

5.1 Partnership with LAUSD

Though the development of the ECS course underwent (and continues to undergo) a rigorous design process, the accompanying policy strategy was essential for ensuring the integration of this course in the school curriculum. New courses require the support and buy-in of school and district administrators. Though each school system is unique, the process of building an accessible program of computer science education in Los Angeles high schools provides lessons and considerations for other school systems interested in reforming computer science education. Due to the interconnected and bureaucratic nature of state, district, and school policies, these efforts require a strategic multitiered approach to integrate this curriculum into the classroom setting so it is accessible to all students.

The first step after completing the curriculum was gaining the buy-in of the school district to offer this course. Armed with the university support to offer professional development and ongoing instructional support in addition to free research-based curricular materials, the Los Angeles Unified School District utilized its policy influence and professional networks to bring the course content to classrooms across Los Angeles.

Importantly, one of the Computer Science Equity Alliance's leaders is the former LAUSD Director of Secondary Science and current school principal of one of the largest LAUSD high schools (which serves the largest number of African Americans in the district), and another Alliance member is now the secondary science director for LAUSD; together they are able to draw on an extensive professional network to offer ECS to principals and place the course and in a growing number of schools. Schools with the highest number of African American and Latino students were especially recruited to offer this course.

The district-level reform initiative in computer science was well supported due to the alignment with other inquiry-based science reform efforts in the district. Because of this alignment, district curriculum officials endorsed this course and the curricular materials. The administrative structure of the school district relies on using the school board-approved memorandum to superintendents and principals in order to formally provide any notices in the district. To publicize this course and gain principal approval, school district Equity Alliance partners work internally to navigate this process and develop informative memorandums about the course and accompanying professional development opportunities. Without this insider status, maneuvering within this complex system would be difficult given the tight resources and multiple demands of school districts.

These official district memorandums were followed up with individual meetings and conference calls with high school principals. Principals who express a signed commitment to offering ECS were asked to identify a dynamic teacher able to attend the extensive accompanying professional development. Since the primary mission of the Computer Science Equity Alliance is to provide greater access to computer science to underrepresented students, we strategically targeted schools serving large numbers of African American and Latino students to offer the course. To solicit their interest, our Alliance team contacted principals by phone and set up meetings to describe the course and professional development. In order to recruit more students of color into computing courses, it was necessary to work with the counselors in schools enrolling this student population.

Once connected with the CSEA community, teachers are provided support and resources (counselor information forms, student recruitment forms, parent information, posters, etc.) to further advocate for computing courses at their school site. Teachers are in an important position for working with counseling offices and other educators to advocate for the steering of traditionally underrepresented students towards enrolling in the ECS course.

5.2 Acquiring College-Preparatory Status for Exploring Computer Science

In 2008, California released a call for the development of new college-preparatory courses that merge the topics of traditionally academic courses with the applied nature of Career Technical Education (CTE). University education researchers collaborated with district officials to submit an application to the University of California Office of the President to deem ECS a college-preparatory elective course. The first year there was a question about prerequisites for the course and the district was encouraged to resubmit the application with a preparatory course in place—a full year later. The resubmitted application reflected Algebra I as the pre-requisite for this course. The University of California Office of the President has approved this college-preparatory application with glowing reviews, noting that this course is “a very strong course submission” and that they would like “to use it as a model CTE course”. This means that the course satisfies multiple purposes for students on various pathways towards college, careers, or other directions.

This state-level and district-level policy strategy ensures buy-in, and more important, sustainability of this course. Though the curricular and professional development support is necessary to develop new courses, the integrated nature of ECS within other district and state reform efforts creates a much more sustainable course. This institutionalization helps with sustainability in the long term.

We have found that our collaborative effort of educational researchers, university computer scientists, district officials, and teachers has been essential to this effort to reform computer science. Securing college-admissions credit approval for a new computer science course in the nation’s second largest school district, and in the largest state, requires a careful navigation of school system demands and opportunities. Each policy move was a result of collaborative decision making of district officials, educational researchers, university computer scientists, professional organizations, and lead teachers. The Computer Science Teachers Association advises this project and situates it within a national context of computer science education. LAUSD district partners were critical proponents; without them the curriculum could not be established within the District. University computer science educators were instrumental in providing content guidance while educational researchers brought in a framework for curricular reform. In addition to the value of multiple perspectives on policy decisions, several of the endorsement applications require district-university partnerships. The strength and successes of the Equity Alliance are a testimony of the power of collaboration. However, there are limitations to shaping only district and state stances on computing. National policy which incorporates computing into education legislation is a necessary step in encouraging state and district leaders to support computing education and intensify the impact and sustainability of local reform efforts [Wilson and Harsha 2009].

5.3 State and National Changes Needed

Unlike science and mathematics, computer science is not considered a “core academic subject” under federal legislation, and thus does not have a place in the core academic curriculum. As a result, computer science often lacks a home department, collegial support, professional development, funding, or required knowledge for students. Still, while many high schools offer a range of computing courses, research has found that most of the knowledge available to students focused on basic computer literacy skills and not the educational standards set by the ACM Model Curriculum for K-12 teachers [Tucker et al. 2003]. The difficulty of local reform efforts is that they must work within the limitations of federal and state policies that fail to support computer science education.

5.3.1 Certification. Perhaps the most pervasive constraint we encounter in Los Angeles is the issue of certification. In California, like many states, there is no computer science teaching certification available. Instead, prospective educators interested in teaching computer science must work towards a business credential or a mathematics credential, and then risk not gaining an assignment teaching a computing class. The lack of certification for computer science teachers also results in a lack of preparation in appropriate computing curriculum and teaching methodology. None of the Los Angeles teachers we work with has ever had a course or any opportunity to develop pedagogical content knowledge in computer science. Many of the teachers are unfamiliar with computer science topics beyond Java programming. This creates many difficulties. As the CSTA notes.

"Where there is no system of computer science certification or endorsement in place, teachers with little or no computer science training are also frequently assigned to teach computer science courses. Without these teachers, many students would have no opportunity to learn computer science. But the continual struggle to stay one step ahead of the students in a constantly changing discipline takes an enormous toll on even the most dedicated educators" [CSTA Teacher Certification Task Force 2008; p. 12].

Though our professional development institutes, workshops, and coaching program are supporting teachers as they teach computer science, it is a challenge for these professional development efforts not to assume any prior engagement with instructional approaches in teaching computer science. For many teachers, this is their first computer science course and they have no prior experiences considering curricular or pedagogical issues in computing education. Certainly, this is the typical case nationwide. Until a national movement towards addressing the certification crisis in computer science education takes place, local reform efforts will continue to have to tackle these issues on their own.

5.3.2 Building a Pathway of Courses. As the College Board reconsiders the APCS programs, there has been much discussion about a sequence of courses in computing. The decision to offer a survey course of major computing topics as a high school computing course was based on a desire to introduce students to the broad field of computing through an exploration of engaging and accessible topics. The curricular materials offer structured activities and projects that allow students to dabble in the computational thinking practices championed by computer scientists [Wing 2006]. It is anticipated that some students will use this course as a launching pad to a soon-to-be revised APCS program.

Though curricular development is most effective when created as part of a local effort, many districts and states do not have the capacity to design their own curricular materials. We believe that the ECS curricular materials could certainly be adapted nationwide to supplement other instructional materials. Apart from the cost of robots, this course does not rely on expensive computers, a particular platform, proprietary software, or costly textbooks—making it an attractive option for schools, districts, and states struggling to provide accessibility with scarce resources. The Exploring Computer Science instructor's guide is available to the computer science education community through the ECS Web site [Goode and Chapman 2009].

The dynamic nature of computer science curriculum, especially in a foundational course, requires a regular cycle of updates and modifications. As software and hardware changes, Web sites are modified, and applied topics in computing shift, curriculum writers must revisit materials on a regular basis. The sustainability of this course relies on an annual revisiting of the ECS curricular materials. Otherwise, it is easy to imagine the course would be outdated in several years. Without a common sequence of courses across schools and states, there will continue to be a dearth of materials for educators trying to create something to meet their school's unique needs.

5.4 Shifting Teacher Belief Systems Takes Time

Our preliminary observations in ECS classroom and subsequent interviews with teachers highlight the difficulty in shifting teaching practices and belief systems in a short period of time. Because ECS is a new curriculum, most teachers are teaching it for the first time, and many of have no prep time built into their day. Though some teachers already incorporated this student-centered teaching in other courses and seamlessly use inquiry-based teaching practices in ECS, others are struggling to understand how (and why) their traditional teaching practices might adapt to improve student learning. Yet, teachers who have little time to learn and reflect on a new curriculum often fall back on established patterns and comfort zones. Teachers are further constrained by testing mandates, prior and more traditional training, shortage of materials, and schools in crisis. The challenge of most inquiry-based reforms for teachers is to teach and test in new ways despite these constraints. Educational scholarship suggests that helping teachers enact new ways of instruction requires nothing short of a fundamental shift in teacher belief systems.

Most national reform efforts suggest that inquiry is left to classroom teachers to shape within the local context. In a sense, the centrality of teachers in implementing inquiry-based practices cannot be understated. Yet, there is little research about how classroom teachers, in diverse classroom settings, enact inquiry in ways that lead to student understanding. Scholars suggest that inquiry research should focus on teacher knowledge, actions, and meanings for inquiry-based science at the center of reform efforts [Keys and Bryan 2001]. Researchers also note the need for our analysis of inquiry to include collaboratively developed inquiry materials on student science learning.

The literature on teacher thinking indicates that, rather than passive recipients of curriculum, teachers are active curriculum creators who make instructional decisions based on a complex system of knowledge and beliefs [Clandinin and Connelly 1992]. Even with a district-endorsed curriculum in hand, each teachers' implementation of the materials relies on their knowledge of the content, beliefs about how students learn and what they can learn, and the structural school context which impacts instructional time and space. The literature suggests that curricular reforms are shaped and altered by teachers' beliefs and understandings of the local context [Keys and Bryan 2001]. In other words, there is an integral relationship between teacher beliefs and the resulting instructional practices.

Similarly, researchers have found that teacher beliefs about the nature of science as an impartial body of knowledge created by a "scientific method" impede inquiry-based teaching [Brickhouse 1990; Gallagher 1991]. To demonstrate this phenomenon, Hashweh categorized science teachers as learning constructivists, learning empiricists, knowledge constructivists, and knowledge empiricists [Hashweh 1996]. His study found that differences in epistemological beliefs influenced pedagogy. The empiricists did not recognize students' prior knowledge, relied on reinforcement as a common teaching practice, and emphasized the scientific method as a framework for instruction. In contrast, the constructivist teacher utilized students' prior knowledge and used a variety of instructional strategies to build on this knowledge and connect new understandings. In our work with ECS teachers, we have witnessed this whole array of different belief systems and teaching practices.

This body of research about how teachers' deep belief systems impact the enacted curriculum serves as an important lesson for reform initiatives that adopt an inquiry-based approach. Providing a curriculum and one-time professional developments might introduce the conceptual approach to teachers, but until teacher knowledge and belief systems are challenged and altered, instructional practices will not necessarily reflect the intentions of curriculum developers and reform leaders. Of course, some teachers already implement an inquiry-based approach in their classrooms, and other teachers are eager to embrace these new ideas. Still, more expansive changes in teacher belief systems require a support structure and reward mechanism that bolsters the goals of reform initiatives.

With a sense of the importance of teacher learning on this project, our design of the ECS professional development program includes and also goes far beyond one-time teacher workshops. Coaching programs and inquiry groups provide teachers other spaces to focus on their instruction and knowledge of teaching ECS. We have learned that a comprehensive professional development program must be deep, personalized, take multiple forms, and be longitudinal in nature to impact teaching.

6. CONCLUSION: THE NEED FOR DEEP ROOTS AND A STRONG FOUNDATION

While there has been much research about the need to transform K-16 computer science education and broaden participation in computing, nationwide we have very few examples of projects that have successfully secured computer science a place in our precollege educational system. To do so requires curriculum transformation, teacher certification and professional development, and policy changes.

Currently in our case study of a partnership with LAUSD, we are wrestling with the tension between: (a) the pace—the steady, arduous work it takes to build this program in a large school district to make it successful and institutionalize it, and (b) the desire—to scale it up, increase the numbers of classes, students, and teachers. While we want to expand the numbers as much as possible as fast as possible, we must also assure that our program (curriculum and professional development program) is solid. This requires additional iterations and revisions.

So, revising the curriculum and our model of teacher professional development may take more rounds than we had anticipated. For instance, in our initial pilot programs, we have learned more about the multiple dimensions involved with bringing inquiry-based learning into the traditional classrooms than we had expected. The reality is that within the world of public education (again, this is where the majority of students are!), things are constantly changing and/or moving at a different pace than we would like. Think: one step forward, two steps back; find a great teacher, teacher gets moved to another class; find a great school, school has budget crisis and must cancel class; have a great professional development and discover that teachers have no prep periods during the day and lacking this time to prepare and practice this new skill, revert to their previously known ways when they return to the classroom, etc. Since there are no blueprints for this type of innovation, we are deeply engaged in a learning-by-doing process and all that entails.

So, there is a danger of wanting to expand our program before there is a rock solid foundation, biting off too much, and spreading ourselves too thin. We know that this is something that many wish was not the case. Indeed, government and industry, for example, are feeling their own set of pressures to reverse the decline of numbers of students studying computer science. But, the educational crisis in this country has created its own set of constraints that must be reckoned with. Building authentic university/K-12 partnerships (and that is the only way to create sustainable change) requires relationships of trust that develop over time.

Though some of the particular details of reforming computer science education outlined in this article are in some ways unique to Los Angeles schools, we hope that the reform model and strategic choices used in this setting will provide guidance for other district-university alliances. Considering how career and educational opportunities in this country are being transformed by technology and computer science, "democratizing computer science" is of critical importance for our nation's education and equity agenda.

REFERENCES

BELL, T., FELLOWS, M., AND WITTEN, I. 2009. *Computer Science Unplugged!* New Zealand.

BRANSFORD, J., BROWN, A. L., AND COCKING, R. R. 1999. *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C.

BRICKHOUSE, N. 1990. Teachers' beliefs about the nature of science and their relationship to classroom practice. *J. Teacher Educ. 41*, 3, 53.

CLANDININ, D. AND CONNELLY, F. 1992. Teacher as curriculum maker. In *Handbook of Research on Curriculum*. 363–401.

CSTA TEACHER CERTIFICATION TASK FORCE. 2008. *Ensuring Exemplary Teaching in an Essential Discipline: Addressing the Crisis in Computer Science Teacher Certification*. Computer Science Teachers Association, New York.

DEWEY, J. 1938. *Experience and Education*. Touchstone, New York.

EGLASH, R., BENNETT, A., O DONNELL, C., JENNINGS, S., AND CINTORINO, M. 2006. Culturally situated design tools: Ethnocomputing from field site to classroom. *Amer. Anthrop. 108*, 2, 347–362.

GALLAGHER, J. 1991. Prospective and practicing secondary school science teachers. *Sci. Educ. 75*, 1, 121–133.

GOODE, J. 2007. If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. *J. Educ. Comput. Res. 36*, 1, 65–88.

GOODE, J. AND CHAPMAN, G. 2009. *Exploring Computer Science*. Computer Science Equity Alliance, Los Angeles. http://www.exploringcs.org/.

GOODE, J., ESTRELLA, R., AND MARGOLIS, J. 2006. *Lost in Translation: Gender and High School Computer Science*. MIT Press, Cambridge, MA.

HASHWEH, M. 1996. Effects of science teachers' epistemological beliefs in teaching. *J. Res. Sci. Teach. 33*, 1, 47–63.

KEYS, C. AND BRYAN, L. 2001. Coconstructing inquiry-based science with teachers: Essential research for lasting reform. *J. Res. Sci. Teach. 38*, 6, 631–645.

KNIGHT, J. 2007. *Instructional Coaching: A Partnership Approach to Improving Instruction*. Corwin Press, Thousand Oaks, CA.

KNIGHT, J. 2008. *Coaching: Approaches and Perspectives*. Corwin Press, Thousand Oaks, CA.

MALONEY, J. H., PEPPLER, K., KAFAI, Y., RESNICK, M., AND RUSK, N. 2008. Programming by choice: Urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM

MARGOLIS, J., ESTRELLA, R., GOODE, J., HOLME, J. J., AND NAO, K. 2008. *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press, Cambridge, MA.

MOLL, L., AMANTI, C., NEFF, D., AND GONZALEZ, N. 1992. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory Practice 31*, 2, 132–141.

MOSES, R. AND COBB, C. 2001. *Radical Equations: Math Literacy and Civil Rights*. Beacon Press, Boston, MA.

TUCKER, A., DEEK, F., JONES, J., MCCOWAN, D., STEPHENSON, C., AND VERNO, A. 2003. *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum*. Association for Computing Machinery, Computer Science Teachers Association, New York.

VYGOTSKY, L. 1978. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA.

WHITE, B. Y. AND FREDRICKSON, J. R. 1997. *The ThinkerTools Inquiry Project: Making Scientific Inquiry Accessible to Students*. Center for Performance Assessment, Educational Testing Service, Princeton, NJ.

WHITE, B. Y. AND FREDERIKSEN, J. R. 1998. Inquiry, modeling, and metacognition: Making science accessible to all students. *Cogn. Instruct. 16*, 1, 3–118.

WILSON, C. AND HARSHA, P. 2009. The long road to computer science education reform. *Comm. ACM 52*, 9, 33–35.

WING, J. M. 2006. Computational thinking. *Comm. ACM 49*, 3, 33–35.

Received February 2010; revised July 2010; accepted November 2010