J. EDUCATIONAL COMPUTING RESEARCH, Vol. 36(1) 65-88, 2007

IF YOU BUILD TEACHERS, WILL STUDENTS COME? THE ROLE OF TEACHERS IN BROADENING COMPUTER SCIENCE LEARNING FOR URBAN YOUTH*

JOANNA GOODE
University of Oregon

ABSTRACT

Despite the digital saturation of today's youth across demographic groups, students of color and females remain severely underrepresented in computer science. Reporting on a sequential mixed methods study, this article explores the ways that high school computer science teachers can act as change agents to broaden the participation in computing for historically underrepresented students. Three high school case studies reveal a critical need for professional development and support to do this work. The subsequent part of the study focuses on the impact of a district-university intervention which trained 25 urban teachers to teach Advanced Placement computer science in their schools. The swift success of this intervention was evident from the following years' dramatic increase in course offerings and enrollment of females, Latinos, and African Americans.

As those with an influential educator in their own past realize, the advocacy and care of a single teacher can have a lasting impact on a student's life interests and pursuits. Yet, despite a general decline in computer science enrollment, a growing

*This work is under the auspices of two NSF-funded grants: "Out of the Loop: Why are so few underrepresented minority high school students learning computer science?" (EIA-0090043) and "Constructing the Computer Science Pipeline: How high school structures and norms narrow access to computer science for underrepresented minority students" (EIA-0213662).

shortage of computer scientists to fill future jobs, and an ongoing segregation of the field, little attention has been paid to the capacity of teachers to reverse these worrisome trends and attract more students to study computer science. In this article, I argue that efforts to increase the numbers of students of color and females studying computer science must begin with the preparation and support of teachers.

Research describes how unequal course offerings between schools, the tracking of only certain students into computing courses, and a chilly climate for females and non-Asian minorities who do enroll in computer science all contribute to the ongoing segregation of the field (Goode, Estrella, & Margolis, 2006; Margolis et al., 2003; Schofield, 1995). The study discussed in this article examines the experiences of urban computer science teachers and explores how teachers can serve as change agents to broaden participation in computing.

This NSF-funded study commenced with case studies of three Los Angeles high schools to examine the experiences and challenges of computer science teachers situated within the context of the school culture. In response to data showing that teachers were lacking content and pedagogical training, as well as an awareness of the many facets of underrepresentation, a pilot intervention program for 25 high school teachers offered professional development to address the teaching challenges which emerged from these findings. This article discusses the effectiveness of the intervention in broadening student participation in computing throughout the district. Longitudinal quantitative enrollment data and follow-up qualitative interviews with eight of the professional development participants suggest successful preliminary results.

LITERATURE REVIEW

A Review of Computing Curriculum in High Schools

Certainly, most educators acknowledge the importance of computing education in schools and flaunt school-wide learning goals along the lines of "providing students with the technological literacy needed for the academic and career challenges of the 21st century." Schools increasingly enforce technology graduation requirements and encourage students to gain critical academic and career technology skills in computer classes. However, a closer examination reveals that all computer courses are not created equal. On one end of the spectrum, computer literacy courses provide students with basic digital skills with a vocational focus. In the middle of the continuum, students in video editing and Web development courses learn digital manipulation and layout skills. On the other side of the spectrum is Advanced Placement computer science, a course designed for students to learn fundamental computational thinking skills and design principles in a problem-solving context. Despite this difference, prior research has found that the title "computer science" is often erroneously used in schools to describe even low-level computer literacy instruction (Goode et al., 2006; Margolis et al., 2003).

What is Computer Science?

In 2001, the two leading professional computer science organizations came together to provide curricular guidance to undergraduate computer science programs (The Joint Task Force on Computing Curricula: IEEE Computing Society & Association for Computing Machinery, 2001). In their report, the task force reviews the curriculum and identifies 14 distinct areas which constitute the computer science body of knowledge: discrete structures, programming fundamentals, algorithms and complexity, architecture and organization, operation systems, net-centric computing, programming languages, human-computer interaction, graphics and visual computing, intelligent systems, information management, social and professional issues, software engineering, and computational science and numerical methods. Faced with the enormity of each of these computer science branches, the task force emphasizes that the ever-growing field of computer science prevents students from learning about all of the areas. Instead, they recommend that core courses be kept to a minimum, so students can pursue one or more computer science branches in depth. Additionally, they assert that students must go beyond simply learning content knowledge, stating, "All computer science students must learn to integrate theory and practice, to recognize the importance of abstraction, and to appreciate the value of good engineering design" (p. 16). This task force concludes that rapidly changing technical and cultural components of computer science require these curricular standards to be frequently revisited and revised as necessary. Unfortunately, these recommendations are not reflected in the high school computer science curriculum.

A brief review of computer courses offered in California high schools (California Department of Education, 2005) uncovers that the only computing course which consistently encapsulates theoretical topics of computer science is Advanced Placement computer science (APCS). Though the Computer Science Teachers Association curricular guidelines (Tucker et al., 2003) call for substantial computer science learning to occur in other computer courses (i.e., Internet publishing, networking, etc.), research of the computing curriculum in a large urban district reveals that this learning rarely occurs outside of APCS (Margolis et al., 2003). Unlike any other computer course in California, APCS counts as an academic elective for college preparation. It should be noted, however, that despite the advanced nature of the course, the content of APCS is narrow in scope. Instead of providing an introduction to the diverse areas of computer science, as recommended by the professional organizations, APCS focuses solely on programming in Java. This constricted interpretation of computer science can serve to discourage students from taking APCS who might otherwise be excited about studying the creative nature of computing.

A Segregated Technology Curriculum

The literature on the "digital divide" in schools shows that access to rigorous computing experiences is not equally available for all students. This curricular divide falls along race and socio-economic lines. Few students, especially poor, youth of color, have an opportunity to study the imaginative nature of computer science and instead are introduced to computing as a set of finite and formal rules, if they are introduced at all (Margolis et al., 2003). The Teaching, Learning and Computing Survey (Becker, 2000) explored the relationship between teachers' use of computer technology, their pedagogies, and school context across hundreds of schools. Researchers found that while students in low-SES secondary schools often use computers more frequently than their affluent peers, their use was limited to vocational and remedial tasks. In contrast, students in high-SES secondary schools were more likely to use computers for academic purposes such as information processing and presentations. The study showed that teacher pedagogy and expectations played a large role in the ways students interacted with technology. Warschauer's (2000) qualitative study of technology's role in two Hawaiian schools arrives at similar conclusions of the ways computer education works to reproduce socio-economic structures, describing how the low-SES school produced technology-able workers, and how the high-SES school produced technology-able scholars.

An analysis of technology course offerings in California public high schools (Table 1) reveals a curricular digital divide (Goode, 2004). Comparing the two quartiles of schools with the lowest and highest rates of student poverty, it is

Table 1. Percent of Schools Offering Computer Courses by Proportion of Students Qualifying for Free/Reduced Lunches and Proportion of Student Population who are African American or Latino

		Proportion of student population qualifying for free/reduced lunch		Proportion of student population which is African American or Latino	
Course name	Enrollment	Highest quartile	Lowest quartile	Highest quartile	Lowest quartile
Keyboarding	37,179	35.7	27.9	32.3*	26.2
Computer literacy	57,965	45.2	22.7	43.2*	35.4
Computer science	3,282	6.5	10.5	7.9	8.3
Web design	4,448	7.0	12.2**	8.7	9.6
APCS A	2,654	5.2	16.2**	6.1	11.8*
APCS AB	925	.9	6.6**	0	6.1**

Note: *Statistically significant at the $p < .01$ level; **Statistically significant at the $p < .05$ level.

evident that affluent students attend schools which are almost twice as likely to offer computer science and Web design courses, three times as likely to offer APCS, and seven times as likely to offer the more advanced APCS AB course than schools attended by low-income students. In contrast, schools serving the highest concentrations of low-income students offer more keyboarding courses and are almost twice as likely to provide low-level computer literacy courses as schools attended by the most affluent students. Similar trends are evident when analyzing course offerings in schools with the fewest and most students of color.

In part because many students of color attend schools that do not offer APCS, the racial composition of test-takers in California exposes the deep segregation of the field of computing (Figure 1). As compared to California public school students, Asians and Whites are overrepresented as APCS test-takers, while African Americans and Latinos are severely underrepresented. Underrepresented students of color make up a combined 53% of the California student population, but they account for only 7% of test-takers (California Department of Education, 2005; College Entrance Examination Board, 2005).

The scarcity of girls in APCS hints that the lack of diversity is related to factors beyond the availability of course offerings. Only 17% of California test takers and 15% of national test-takers are female, making APCS the most segregated AP field in terms of both race and gender. In fact, out of 2788 California APCS exam-takers, only 1.5% were Latinas and only .2% were African-American females. These are shocking statistics for a minority-majority state which birthed and nurtures Silicon Valley and houses several of the nation's top-notch computer science research universities. This alarming data begs us to further examine the question, "Why are so few students of color and females studying computer science?"

A Chilly Computer Science Classroom

When advanced computer science courses are available, they are often gendered and racialized spaces, even within diverse high schools (Schofield, 1995). An extensive urban high school study exploring students' decisions to enroll or not enroll in computer science revealed several structural dynamics that keep students away from computer science courses (Goode et al., 2006; Margolis et al., 2003). To begin, even though students hold preconceived notions of who studies computer science, few of them are able to articulate what computer science is. Without understandings of computer science, its interdisciplinary connections, and an explicit bearing to youth interests, few students not already enamored with computing end up pursuing these courses. This problem is exasperated by guidance counselors and teachers who act as gatekeepers to computer science courses, steering certain students towards the courses, and directing other qualified students away. Such tracking reflects the biases and beliefs of counselors and teachers, rather than the aspirations and abilities of students (Oakes, 1985, 1990). In computer science classrooms, the composition of the students, the curricular

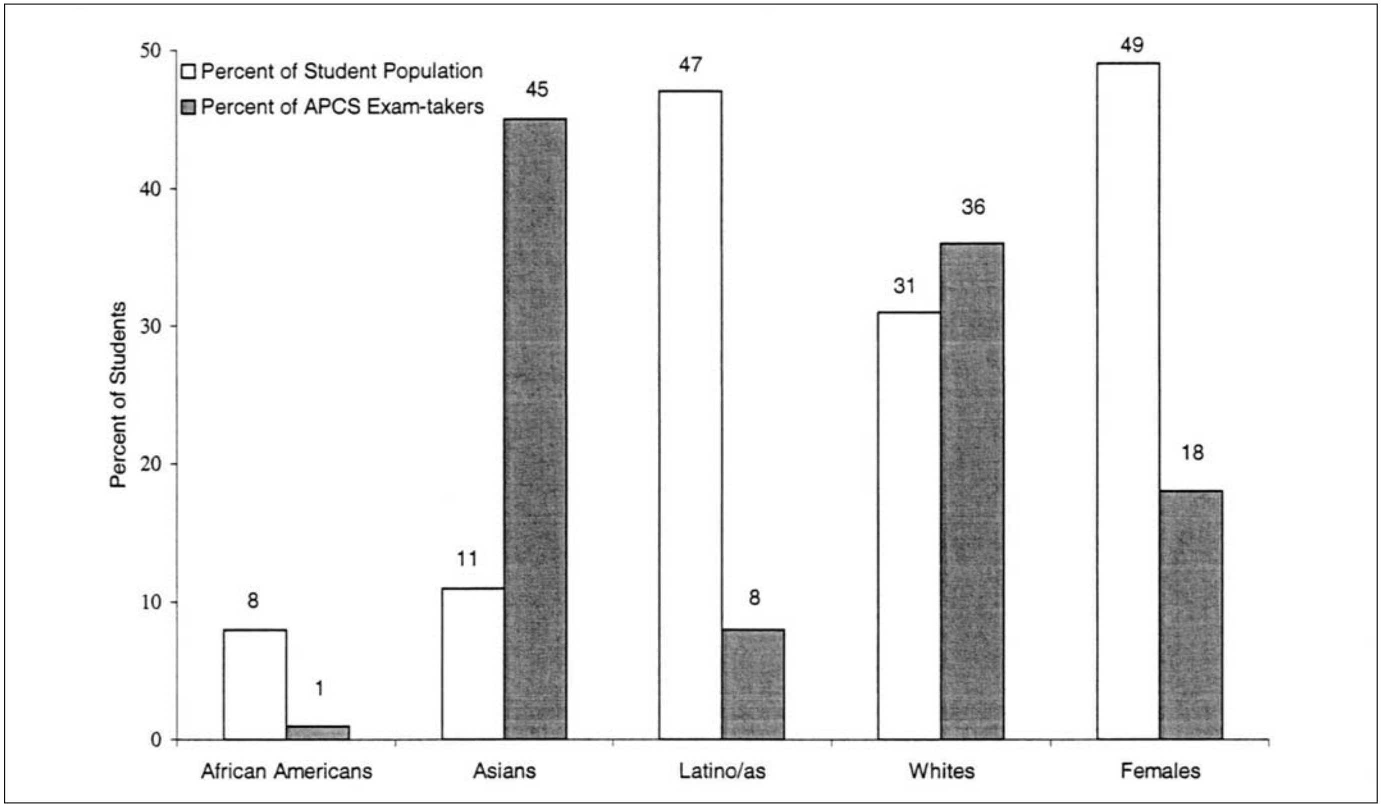

Figure 1. California public school population and APCS exam-takers by ethnicity and gender.

assignments, and teacher pedagogy all contribute to the perpetuation of a chilly climate (Goode et al., 2006).

THEORETICAL PERSPECTIVE

The review of literature highlights the ways that schools contribute to this computing segregation by providing low-level skills and few resources (including qualified teachers and an engaging curriculum) to low-income students and offering more sophisticated types of knowledge for wealthier students. To better account for why so few underrepresented students study computer science and explore ways teachers can broaden participation, this study relies on a theoretical lens that blends concepts of social reproduction with ideas of teachers as influential social change agents.

Computing Opportunities as an Indicator of Social Reproduction

With a growing influence on information and communication technologies, much of the country's future economic growth depends on a strong computer science workforce. The affordances of knowing about computer science are significant in this digital, information-age. Based on the importance of technology on our economy, computer science can be considered, high-status knowledge. As Apple (1990) asserts, "The possession of high-status knowledge, knowledge that is considered of exceptional importance and is connected to the structure of corporate economies, is related to and in fact seems to entail the non-possession by others." In essence, high status knowledge is "by definition scarce, and its scarcity is inextricably linked to its instrumentality" (p. 36).

The depth of the computer science curricular divide indicates a strong relationship between the socio-economic status of the student population and curricular opportunities to engage in a field of high-status knowledge. Social reproduction theorists explain differential curricular offerings between different student populations as a evidence that schools function to preserve the social-economic position of society (Anyon, 1995; Bowles & Gintis, 1976; MacLeod, 1987; Willis, 1981). Anyon's (1981) seminal ethnographic study of five elementary schools serving different socio-economic groups of children illustrates the stratification of curriculum. She found that teaching methods and philosophies of education at each school operate to prepare students to occupy the same position of the labor market as their parents. In computing, this framework can be used to examine the inequities in the types of computing experiences students encounter, including the curriculum and teaching methods. Working class students learn about technology in high school to prepare for working class jobs, while affluent students learn about technology in high school to prepare for college, and eventually, high-status knowledge careers.

Though patterns of social reproduction are difficult to disrupt, addressing issues of opportunities to learn for students can create more equitable allocation of educational resources. In fact, research shows that the most influential resource for students' opportunities to learn and achievement levels is access to qualified teachers (Darling-Hammond, 1997, 1998). Perhaps some of the structural limitations to participation in computing can be addressed and circumvented if teachers are offered the resources to design computer science learning environments which attract and retain more females, African Americans, and Latino students.

Teachers as Powerful Change Agents

If given the tools, the agency of teachers can work to disrupt the organization of schooling that limits opportunities for historically underrepresented students to access high-status knowledge. Though teachers might have some experience as change agents in other arenas, educators might lack the specific skills necessary to narrow the access divide in computer science education. For instance, it is difficult to build a successful computer science program without any knowledge of the field of computer science, or interaction with other computer science teachers. In his article *Why Teachers Must Become Change Agents*, Fullan (1993) notes that though many teachers enter the field with a moral purpose, the ability of teachers to serve as change agents relies on four core capacities: personal vision, inquiry, mastery, and collaboration. Fullan argues that these capacities are not developed individually, but must be nurtured and consciously developed in a professional setting.

Professional development programs can serve as spaces to purposefully develop these essential capacities. In her article considering the alignment of school reform and professional development programs, Little (1993) discusses "streams" of reform efforts. Two of her streams lend relevance in considering of how teacher training programs might help computer science educators develop the capacities of change agents. First, Little describes models of professional development that focus on developing extensive subject matter knowledge, pedagogy that departs from textbook-centered teaching, and an instructional design which engages students in learning. This type of focus can serve to build the mastery of computer science knowledge and develop pedagogical content knowledge (Shulman, 1986, 2000) specific to computer science. The second stream of professional development experiences centers on problems of equity and aims to "assist teachers to identify and alter classroom practices that contribute to failure and that undermine equal opportunity to learn" (Shulman, 2000, p. 131). This type of professional development provides a space for teacher inquiry in considering their own practices that can serve as gatekeepers or motivators in students' interest in computing. This equity-based model of training creates an opportunity for teachers envision their role in addressing the problem of underrepresentation in computing. Both types of professional development programs hold great promise in promoting collaborative activities and encouraging professional social networks between teachers. Among all professional developments programs considered for her study, Little points out that summer institutes sponsored by the NSF where "teachers enjoy sustained work with ideas, materials, and colleagues" (Shulman, 2000, p. 135), were among the most successful and favorable model of training.

Teachers with the capacity of change agents hold considerable promise in reaching out to students in an effort to desegregate the field of computing. If given opportunities to develop mastery of content matter and pedagogical content knowledge, teachers can better introduce students to the dynamic field of computer science. Deeper opportunities to reflect on unequal access and participation can help teachers understand how to create pedagogical practices than draw in, rather than keep out, females and students of color. Broadened collaborative activities allow teachers to know one another and develop professional networks.

Together, the ideas of social reproduction and teachers as change agents help explain the current segregation of computer science and present a pragmatic model of how to support classroom teachers in building sufficient knowledge to diversify their own computer science classrooms. To further probe these ideas, this study considers three questions, "Who teaches computer science?", "What are the computer science teaching conditions in urban schools?", and "With the proper support, what can teachers do to broaden participation in computing in their own schools?"

METHODOLOGY

In order to examine these questions, this article draws from several sources of evidence to provide a multidimensional analysis of the relationship between access to quality computer science educators and opportunities for under-represented students to study computer science. This sequential mixed methods study begins by considering the context of computer science teaching in urban schools and the professional characteristics of California APCS teachers. The second phase of the study examines the impact of a professional development intervention in increasing the numbers of African Americans, Latinos, and females studying APCS.

The study began with a 3-year examination of computer science educators in Los Angeles Unified school district (LAUSD). Three schools were selected for in-depth case studies: East River, a severely overcrowded school serving low-income students, 99% whom are Latino/a; Windward, a school serving primarily middle-class African-American students; and Canyon, a school serving an ethnically and socio-economically diverse population and located in a white, wealthy community. The case studies, while focused on the computing curriculum and teaching conditions at each school, embraced the notion that understanding a single aspect of schooling requires an examination of the general learning conditions within the schools (Oakes, 2002). Thus, school conditions of competing policies and inadequate learning conditions contextualized descriptions of the availability and quality of computer science courses. At each of these three schools, the principal, vice principal, counselor, computing teachers, and mathematics teachers were interviewed. In total, 30 educators participated in the study. Along with regular classroom observations, their interviews provide valuable insight on the intersection of teaching conditions and student opportunity to explore computer science.

To provide a statewide portrait of computer science teachers, results from a quantitative analysis of two datasets published by the California Department of Education are reported. The "Professional Assignment and Information Form" dataset includes information on the gender, race, professional credentials, and teaching experience of all public California teachers. The "Assignment" dataset includes all courses offered in California and was linked with the former dataset to explore teacher characteristics by course assignment. Both datasets are based on information supplied by teachers and schools shortly after the beginning of each school year. Only educators with an APCS teaching assignment in 2004-2005 are included in this demographic summary.

Next, this article draws on data collected from APCS classroom teachers who attended a pilot intervention program that took place in the summer of 2004. In response to the findings from the first phase of this study, an alliance between LAUSD, UCLA's Graduate School of Education and Information Studies, and UCLA's Henry Samueli School of Engineering and Applied Sciences was formed to increase the capacity of the district's teachers. The Computer Science Equity Alliance's first effort was the creation of a professional development program for LAUSD teachers who already taught, or could potentially teach, APCS. This 2-week intervention aimed to build teachers' knowledge in four ways: a) deepen teachers' content knowledge of Java and APCS-specific content; b) develop teachers' pedagogical strategies appropriate for urban youth and girls; c) increase teachers' awareness of the issues related to the underrepresentation of Latinos, African Americans, and girls in computer science; and d) provide teachers a space for engaging in collaborative inquiry and building professional networks of support with their colleagues from different schools.

Teachers were recruited via a district memo sent to school administrators instructing principals to identify potential APCS teachers. As a result, 21 LAUSD high school principals recommended 25 teachers to attend a 2-week summer professional development program. Four of these teachers were already APCS teachers but desired additional support from the district. All teachers agreed to participate in this research study. Participants wrote professional technobiographies outlining their own history, knowledge, and beliefs about teaching and learning computer science. Eight teachers who taught APCS the subsequent year consented to follow-up classroom visits, interviews, and several e-mail exchanges to probe their emerging experiences as APCS teachers. To determine the impact of this pilot program on student enrollment patterns, this study included an analysis of LAUSD APCS enrollment changes between the 2003-2004, 2004-2005, and 2005-2006 school years. For comparison purposes, statewide APCS enrollment is also reported to consider the effects of this district program.

The collective results of these complementary sources of data shed light on the relationship between the availability of highly prepared and supported computer science educators and student participation in computer science.

FINDINGS

A Computer Science Teacher Shortage

The computer science teachers at the three case study schools are certified in mathematics, social studies, and business. The variety of subject area backgrounds is typical of the state's computer science teachers since the California Commission on Teacher Credentialing does not offer a teaching certification specific to computer science. Though computer science is designated as a business course by the state of California, only 17% of APCS teachers hold a credential in that area. That means that a full 83% of APCS teachers in California, most of who are mathematics teachers (see Table 2), are technically teaching out of subject area. However, federal education policy that mandates appropriate subject-area certification does not apply to non-core academic subjects like computer science.

Table 2. Credentials Held by California AP Computer Science Teachers

Credential type	Percent of APCS teachers holding credential*
Mathematics	66
Physical sciences	18
Business	17
Industrial technology	16
Social sciences	16
Life sciences	6
Vocational education	6
English	5
Foreign language	4
Physical education	3
Art	2
Music	2

Note: *Totals add up to more than 100% due to teachers holding multiple credentials.

The credentialed areas of the 25 teachers participating in the professional development program closely match the statewide proportions in terms of credentialed subject areas. Additionally, analysis of their techno-autobiographies indicates that this pool of LAUSD teachers represents a broad range of prior computer science experience and formal education. Among the participants, 4 teachers majored or minored in computer science; 6 majored in mathematics; 9 majored in business; and the other teachers received their degrees in linguistics, psychology, music, social studies, Spanish, and English literature. Their experience with computer science varied from no familiarity at all to teaching APCS for almost 2 decades. These results show that there is no clear path to becoming a computer science teacher, leading to a shortage of knowledgeable computer science educators.

Without a Home: Unique Challenges of Teaching Computer Science in Urban Schools

Observations and teacher interviews at East River, Windward, and Canyon expose the effect of computer science teachers not having a home department. Since most schools offer only a couple of computing courses taught by the same teacher, the individual instructor must act as pseudo computer science department chair to advocate for the placement of the course on the school's master schedule, gain computer lab access, install necessary software, support the hardware, and secure curriculum. These teachers also put in additional effort into advertising the course and recruiting students. However, the inability to obtain necessary computer science departmental educational resources creates difficulties when funding is often delegated directly to core academic subjects. At East River, Matthew spoke with frustration about maintaining the antiquated equipment necessary for his programming class with little financial support. "I know a lot of the things I've asked for, like a disk drive or something, the mathematics department has provided. The technology coordinator has not provided anything. But I would like to have more support from him, money, to be able to buy things or whatever." The school has never provided software or textbooks for this programming course, leaving the struggling teacher to develop and deliver all curricular content on his own.

At Windward, a technology coordinator described the lack of curriculum for computer science courses. "I don't know if there's any curriculum out there that's already set up [for computer science], we have to kind of do that ourselves 'cause there's no state guidelines for business, and computer science is put with the business department, even though it's college level." Samuel, a Windward computing teacher without a strong technology background, believed that textbooks would be an integral support for him and students but the school did not have funding for this elective course. Thus, the teacher (against school policy) encouraged students to buy their own textbooks for classwork. In contrast, at

Canyon, students in all the computer science courses were provided textbooks for their academic learning.

Findings also reveal how the absence of professional computer science colleagues at a school site precludes collaborative lesson planning and a shared community of knowledge with other faculty, further isolating computer science teachers. Before the intervention, none of the computer science teachers we talked to interacted with computing teachers from other schools. Never had computer science teachers been provided professional development by the district, or even come together to meet to discuss issues pressing to computer science education in LAUSD. Still, especially at East River and Windward, it was painfully clear that these teachers could use professional support.

Insufficient Teacher Preparation

At East River, the most advanced computer science course, programming, is taught by a mathematics teacher who inherited the course when the prior teacher took on department chair responsibilities. Matthew reports that he struggles to stay ahead of his students, and has mostly taught himself the *Basic* language at home on a pre-Windows machine. The department chair, who has worked at the school for over 20 years, notes the vacuity of computer science educators, stating:

> In high school there's not enough [computer scientist teachers] so they try to link it in with math, but math teachers don't necessarily come out with computer science backgrounds so you don't have the people to teach it. And that's the big thing. We don't have the people. Matthew never taught it. He's picked it up. Where did he get it from? He got it from me teaching him, going through programming on the TI83. That's where he started. Everything is self taught. No one wants to put that time in. Very, very few teachers will do that. He happens to be one of the few.

Despite his good intentions and long hours of preparation, Matthew's lack of computer science knowledge resulted in dull assignments, such as state capitol trivia games, which failed to spark the interests of the students in the course, especially the girls.

The technological portrait of Windward includes a similar story of ill-prepared computer science teachers. At this school, the most advanced computing course was assigned to a first-year teacher with a background in social studies. Samuel's interest in computer graphics encouraged him to develop a learning environment in which students created visual design projects. Still, little, if any, more sophisticated computing topics are introduced in this classroom. The head counselor at Windward echoes the comments made by East River's mathematics department chair regarding the importance of teacher knowledge and availability. He points out that "it's not only getting students prepared, it's having teachers available to teach those classes and getting to offer those classes because we only get a certain number of teachers and first you're going to offer basics."

Canyon's computing curricular offerings are much more cohesive and advanced. In this affluent school, there is an established computer science curriculum that leads to APCS. The courses are taught by Katherine, a late-entry teacher who worked as a programmer in her prior career and has extensive knowledge of several programming languages. Still, despite her computer science skills, classroom observations uncovered how her pedagogical approaches often work to the detriment of the handful of females and students of color in the class.

A Research-Based Intervention

The pilot intervention was designed to go beyond providing teachers with computer science content knowledge; rather, it intended to proactively address issues of isolation, curricular resources, technical considerations, counselors, pedagogy, and other topics that can create obstacles to teaching computer science. The program provided space to develop all the tools of change agents: inquiry, personal vision, mastery, and collaboration. Readings and discussions considered the historical segregation of computer science and specific ways in which teachers might better recruit and retain underrepresented students. Teachers had opportunities to consider alternative teaching methods, such as role playing, to explore a more engaging curriculum. Importantly, teachers worked together in small groups or pairs throughout the program, building professional relationships and learning from one another.

An important part of the intervention was to consider computer science topics besides programming as a way of "hooking" students into studying computer science. Teachers toured several university laboratories demonstrating computer science research and applications. The high school teachers watched software mimicking precise human movement with the effect of gravity, witnessed the use of interactive set props in the theatre department, and observed environmental scientists collecting data using censors with attached radios. Teachers reported that they were excited to witness the emerging field of computer science, and, as one teacher commented, he found them to be "something inspirational I will take to my classroom." But more importantly, these teachers experienced a more holistic picture of computer science. As another teacher reported, these tours "made me realize for the first time that computer science is more than Java programming."

Professional Networks and Curricular Change

An important finding of the continuing impact of this intervention was the development of a professional network between the teachers. Reflecting on their experiences in the intervention program, the eight interviewed teachers all emphasized the importance of their new collegial relationships with other computer science teachers. For the first time, LAUSD teachers were given the space to meet as a computer science department. As a result, though only one of these teachers works at a school with another knowledgeable computer science educator, none of the intervention participants reported school-level isolation as a problem, due to their relationships with computing teachers across the district.

This critical mass of computer science teachers has also led to increased power in drawing attention to their common challenges. For example, the curriculum the district initially provided in the training ended up being a poor fit for APCS students. The teachers overwhelmingly expressed frustration about using a curriculum not designed for teaching youth the subset of Java and case study as outlined by the College Board's list of course topics. Rather, this resource provided online modules for adults to learn the programming language of Java. The teachers reported that the curriculum was "too dense," "too high level," and "not friendly" for students. So, many of these teachers abandoned this curriculum, and made use of their new professional networks with other APCS teachers to pool other curricular resources. In the meantime, educational researchers synthesized these teacher narratives for district officials, and recommended an APCS-focused curriculum. Soon, the new curriculum had been approved by the district. As individual teachers, it would be unlikely a request for new curriculum would be considered, but, with the voices of many teachers providing feedback to the district, their ability to influence change is increased.

Needed: Additional Teacher and Student Support

After the intervention, teachers expressed a concern of their ability to teach more advanced APCS concepts. The 2-week initial Java instruction at the professional development was insufficient to master the content of the Java subset for the APCS course as well as the "Marine Biology Simulation case study" which accounts for 25% of APCS exam questions. Though successfully engaging their students in the introductory computer science ideas, these teachers advocated for further training opportunities to support their teaching endeavors. This was especially true amongst the teachers with the least prior computer science knowledge. For the most part, the teachers had felt confident launching the course, but many reported that they were largely trying "to stay one step ahead of students" as they encountered more difficult topics.

In addition, the teachers reported a desire for increased student support from the university. They pointed out that many of these students need extra tutoring, or an expanded view of computer science, but knew no adults besides their teacher who could help them understand these challenging college-level topics. In response, UCLA and LAUSD began offering monthly AP Readiness classes at UCLA. These Saturday sessions provide additional support and networking opportunities for teachers in tandem with supplemental instruction for students in preparation for the APCS exam.

Preserving APCS in Urban Schools

This study also reveals how competing federal and state policies can serve to limit opportunities to offer computer science courses, especially in low-achieving schools under extra scrutiny to provide "the basics" for students. Interviews with administrators at the case study schools, for example, reinforced the marginalization of computer science in schools plagued with low resources and high external demands. At East River, the principal and counselor both reported that unless computer science could lead to higher test scores, then they were unable to devote any resources to support a course.

But, even after committing a teacher to the intervention program and creating a successful APCS presence on campus, the preservation of computer science courses is still at risk. The story of a talented APCS teacher presents a sobering reminder of the urban schooling context of this study. As a programming major in India, Prabha attended the intervention to learn more about the APCS course and develop a program at her school, located in the impoverished area of South Los Angeles. Her recruitment efforts were effective, as she had attracted 2 African-American males, 2 Latino males, and 14 Latinas into her class (in the prior year, only 19 Latinas in all of California took the APCS exam). Observations throughout the year showed a rigorous level of teaching and learning inside her classroom. Still, the success of Prabha's APCS course could not be maintained in a school environment plagued with low achievement and scarce resources. With only 32% of the high school students passing the mathematics portion of California's high school exit exam, the principal increased the number of remedial mathematics courses, and in following *No Child Left Behind* mandates, needed to staff these courses with "highly-qualified" mathematics teachers. As a certified mathematics teacher, Prabha's teaching course-load was filled with mathematics courses and left no space for teaching APCS. The course was passed to an unqualified teacher, and ultimately dropped from course offerings.

Intervention Leads to More Courses, Broadened Participation

The research-based design of the intervention built on the hypothesis that increasing the pool of knowledgeable computer science teachers with specific training in attracting and retaining underrepresented students would lead to broadened participation in APCS. An examination of enrollment statistics in APCS the year before and after the intervention confirmed that this intervention was successful in increasing the availability of LAUSD APCS courses; the number of schools offering the course doubled from 11 to 22. This increase of courses in turn expanded the access for students to enroll in APCS at their schools. Before the intervention, only 225 students studied APCS, the year after the intervention, 395 students studied APCS. In the 2005-2006 school year, 23 schools offered the course and 619 students studied APCS—over 2 and a half times as many students as before the intervention. Figure 2 displays the enrollment trends by ethnic group and gender over three years (Los Angeles Unified School District, 2005).

The contrast of these district enrollment increases with statewide enrollment patterns illustrates the impact of the intervention program. Comparing the enrollment the year before and after the intervention displays that while more LAUSD students studied APCS, fewer California students enrolled in the course. In a 2-year period, APCS student enrollment in LAUSD increased by 75%. During the same period, non-LAUSD California student enrollment decreased by 17% (California Department of Education, 2004, 2005). Clearly, it seems these exciting increases in the district can be attributed to the efforts of the pilot intervention program and not other computer science education trends.

Strategic Recruitment

The data indicates that these changes in enrollment cannot be attributed to the mere availability of the course at the school site. Though placing the APCS course on an urban high school's master schedule is both challenging and necessary, teachers are stepping outside of their role as classroom teachers to disrupt the typical APCS enrollment process that has historically not attracted girls, Latinos, or African Americans. Follow-up interviews with participants reveal their agency in purposeful recruiting of students, informing counselors of appropriate students to funnel into the course, and building an exciting, supportive APCS learning environment that builds a reputation that attracts students.

The story of one teacher exemplifies how strategic recruitment on the part of the APCS teacher can have a significant impact on student participation. Disappointed with the small number of students who enrolled in APCS the year after the intervention, this teacher embarked on a recruitment campaign the following spring just before students began enrolling in their fall courses. She wrote letters to all juniors enrolled in Intermediate Algebra, the prerequisite for APCS, personally inviting them to join her class the following fall. She visited mathematics classes and talked about computer science. As a result, in 1 year, her class grew from 12 to 60 students, increasing the number of girls from 0 to 20.

DISCUSSION

The primary purpose of this study was to examine the role of teachers in providing computer science learning opportunities for urban students, especially females, Latinos, and African Americans. Though the first part of this study found that teaching conditions for urban computer science teachers often are void of any resources or support, the intervention shows that if given the necessary training, teachers can increase opportunities for underrepresented students to study computer science. Yet, most districts do not have the capacity of LAUSD and UCLA in

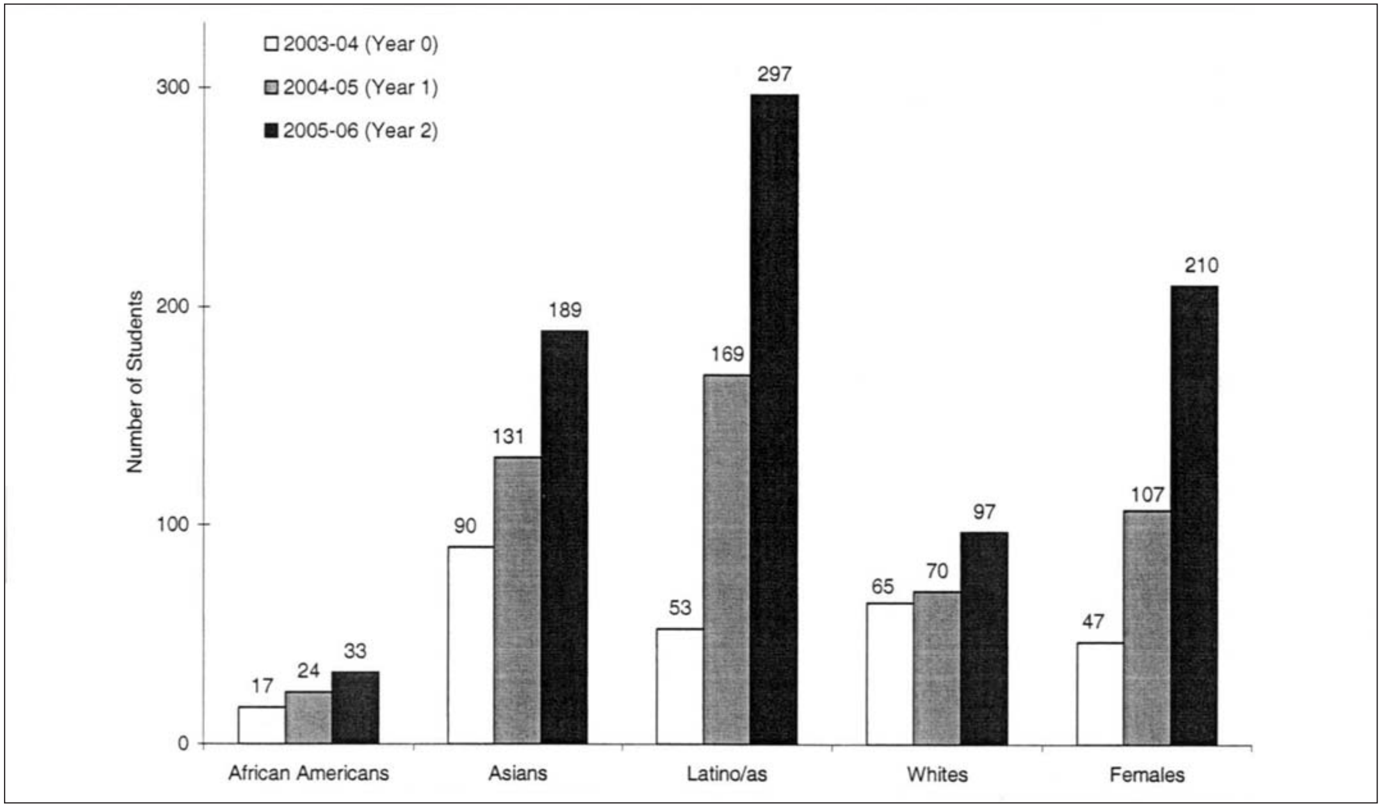

Figure 2. LAUSD APCS Enrollment 2003-2006 by ethnicity and gender.

providing such in-service professional development for teachers. Pre-service training, however, can contribute to the production of a steady and knowledgeable supply of computer science teachers. Only with more prepared computer science teachers will our educational system be able to produce the diverse, technologically-savvy workforce required for our economy's prosperity.

Increasing the Supply of Computer Science Teachers

In contrast with LAUSD, smaller districts are unlikely to have sufficient resources to offer the same level and depth of training that this intervention provided for LAUSD teachers. To more strategically widen the pool of qualified computer science teachers, increased consideration should be given to certification programs that target pre-service teachers, especially those who have studied computer science in college. These preparatory programs can help teachers to master content, explore pedagogical approaches, and importantly, develop professional networks with other computer science teachers.

At this point, it is difficult to provide a national perspective on the availability of computer science teaching certificates which differ on a state-to-state basis. As discussed earlier, California, like many other states, does not offer a computer science teaching credential. Instead, California places computer science as a topic associated with a business teaching credential. In 2003 and 2004, two consecutive surveys distributed by the Computer Science Teachers Association (CSTA) were completed by thousands of secondary computing teachers. Results revealed that approximately half of teachers reported that a teaching certificate for high school computer science was offered by their state (Computer Science Teachers Association Research Committee, 2005). Oddly, when tabulated by state, this split continued, displaying the obscurity of certification in this field. The CSTA is currently surveying state certification offices of credentialing opportunities for computer science teachers (Stephenson, 2005). At this point, it is still unknown how many states provide computer science certification. Once this information is compiled, and a clear picture emerges of computer science certification programs nationwide, policymakers should consider streamlining ways in which teachers gain important computer science knowledge and pedagogy before entering the classroom. This study highlights that increasing the supply of qualified teachers is a critical component in providing increased opportunities to access this high-status knowledge.

Still, though several teacher education programs embody a social justice and equity-based perspective (Cochran-Smith, 2005), the majority of preparatory programs do not. Without careful thought in preparing teachers to become social change agents in computer science education, teachers might not have the knowledge or skills to address the gender and racial gap in the field. In fact, without particular attention to the research examining this digital divide, teachers might learn to teach in a way that continues to attract primarily White and Asian boys, perpetuating the cycle of social reproduction. Thus, though certification programs for computer science hold potential to increase the general supply of computer science teachers, it is critical that these teachers attract underrepresented students into advanced computer science classrooms.

Of course, even with more streamlined pre-service preparation programs for computer science teachers, ongoing in-service professional development programs must continue to be available for practicing teachers. The content of computer science is more prone to change than any other subject in school, requiring constant support for teachers to keep up. For instance, the programming language of the APCS test changed from Pascal to C++ in 1999 and then to Java in 2004. That's three languages in the course of 7 years. Professional development can provide an environment for teachers to revitalize professional networks while keeping up with developments in the APCS course and in the field.

Computer Science Education: A Dual Crisis

The importance of this study needs to be considered in the context of a dual crisis in computer science education—too few students pursuing college degrees in computer science to fill jobs in the economy, and a disproportionately low rate of participation of females, Latinos, and African Americans studying in the field. First, there are simply not enough students pursuing computer science to fill projected jobs. According to the Bureau of Labor Statistics, five of the six fastest growing occupations from 2004-2014 which require college degrees are in the field of computer science, creating a 30.7% increase of jobs in computer and mathematical science occupations (Hecker, 2005). All of these professions are considered "very high" in median annual income. Yet, across the nation, enrollment in computer science classes in high school and college are decreasing. Between 2002 and 2005, APCS test-takers nation-wide has decreased by 19% (College Entrance Examination Board, 2002, 2005). The severity of declining participation is more alarming in higher education. Annual surveys of computer science departments shows that after five years of declining enrollment, the number of new CS majors in 2005 was half of what it was in 2000 (Vegso, 2006). Results from an annual survey of college freshmen confirm a declining interest among incoming students in pursuing a computer science major over the last 5 years: from 3.8% in 1999, to 1.4% in 2004 (Vegso, 2005a).

Alongside this decreasing enrollment, the racial and gender disparities in computer science reveal that there is large amount of untapped talent which could fill critical positions in computer science. Women receive only 15% of bachelor's degrees in computer science, and Latinos and African Americans account for only 8% of undergraduate degrees. An analysis an annual survey of freshman at undergraduate institutions reveals that this problem is actually getting worse—in 2004 the proportion of female freshmen expressing interest in majoring in computer science dropped to the lowest rates since the early 1970s (Vegso, 2005b).

Diversifying computer science is a critical and important task. It is not just an issue of equity for underrepresented groups, but increased representation is vital for the continued growth and development of our economy. With computer scientists designing today the technology we all use tomorrow, it is critical that the inventors reflect the ethnicity, gender, and socio-economic groups that ultimately become the end-users in our diverse society. This current void of diversity in the computer science pipeline narrows the methods and end-products of technology applications. James Gates, a renowned African-American physicist, notes that folks from different cultural groups bring nothing less than a *style* of science with them. In an interview Gates uses a metaphor of music to illustrate the importance of diversity. He explains:

> There are styles of physics that are Russian, Germanic, English, and even American, which is very detectable to me. When enough people of African heritage do physics, they're going to bring a different aesthetic, and it will be new and valuable. Because classical music and jazz exist we don't think that we're musically poorer. Had jazz never come into existence we would've been musically poorer, but before jazz, musicians could say, "We're doing just fine. We have this wonderful art form here." And that's what's lost when people with different inputs don't participate in science. We miss the opportunity to create jazz (Parks, 2003).

As educators, it is time to jazz up computer science.

Yet, while enrollment in computer science courses has decreased across the nation since the turn of the century, and the gender divide has broadened, the overall APCS enrollment in LAUSD has significantly increased over the last 3 years. Additionally, though computer science classrooms have had a perpetual underrepresentation of females and students of color, this pilot intervention has led to increased participation of girls, Latinos, and African Americans studying computer science in LAUSD. The promising results of this intervention demonstrate that given adequate and appropriate professional support and resources, teachers can disrupt the dual crisis in computer science education and entice students into studying computer science. Yet, there are still challenges which must be addressed in future efforts. Though 23 LAUSD high schools now offer APCS, 76 schools still do not offer the course, including East River and Windward. Clearly, though these initial intervention results are encouraging, there is still much work to be done to provide all LAUSD students the opportunity to study computer science.

CONCLUSION

This study explains how teachers can act as social change agents in broadening the participation in computing, especially for students who have been historically underrepresented in the field. Though this mixed methods study is grounded in the context of LAUSD, and thus not generalizable to other communities, the results detail how investing in the preparation of qualified computer science teachers can lead to broadened participation in computing. Thus, future efforts seeking to remedy the dual crises in computer science education should consider the important role of teachers. With more future jobs in computer science than students pursuing degrees in the field, this low participation of students is both an issue of equity and a critical concern for the preparation of our future economic workforce.

Still, teachers cannot advocate for students without training and resources. The challenge in changing the face of computer science requires increased attention and a coherent set of programs aimed at supporting the endeavors of teaching and learning computer science in urban high schools. Like other reform efforts in the public school system, diversifying computer science requires district-wide institutional support as well as a local school-level commitment by administrators, counselors, and teachers in mathematics and science. In conjunction with increased opportunities for females and students of color to enroll in computing courses, educators must be encouraged to broaden their vision of the field of computer science and associated culturally relevant pedagogical practices. With these factors in place, teachers can provide rich learning experiences to encourage underrepresented students in studying computer science.

REFERENCES

Anyon, J. (1981). Social class and school knowledge. *Curriculum Inquiry, 11*(1), 3-41.

Anyon, J. (1995). Social class and the hidden curriculum. *Rereading America,* 45-60.

Apple, M. W. (1990). *Ideology and curriculum* (2nd ed.). New York: Routledge.

Becker, H. J. (2000). *Subject and teacher objectives for computer-using classes by school socio-economic status*. Irvine, CA: CRITO.

Bowles, S., & Gintis, H. (1976). *Schooling in capitalist America: Educational reform and the contradictions of economic life*. New York: Basic Books, Inc.

California Department of Education. (2004). *California Basic Educational Data System.* [On-line]. Available: http://www.cde.ca.gov/ds/sd/cb/

California Department of Education. (2005). *California Basic Educational Data System.* [On-line]. Available: http://www.cde.ca.gov/ds/sd/cb/

Cochran-Smith, M. (2005). The new teacher education: For better or for worse? *Educational Researcher, 34*(7), 3-17.

College Entrance Examination Board. (2002). *Advanced placement program California and national summary reports*. New York, NY: The College Board.

College Entrance Examination Board. (2005). *Advanced placement program California and national summary reports*. New York, NY: The College Board.

Computer Science Teachers Association Research Committee. (2005). *CSTA national secondary computer science survey (2005)*. [On-line]. Available: http://csta.acm.org/Research/sub/survey_pdfs/CSTASurvey05Results.pdf

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

Darling-Hammond, L. (1998). Teachers and teaching: Testing policy hypotheses from a national commission report *Educational Researcher, 27*(1), 5-15.

Fullan, M. (1993). Why teachers must become change agents. *Educational Leadership, 60*(6), 12-17.

Goode, J. (2004). Mind the gap: The digital dimension of college access. *Dissertation Abstracts International, 65*(11), 4166 (UMI No. AAT 3155020).

Goode, J., Estrella, R., & Margolis, J. (2006). Lost in translation: Gender and high school computer science. In W. Aspray & J. M. Cohoon (Eds.), *Women in IT: Reasons on the underrepresentation* (pp. 89-114). Cambridge, MA: MIT Press.

Hecker, D. E. (2005). Occupational employment projections to 2014. *Monthly Labor Review, 128*(11), 70-101.

Little, J. W. (1993). Teachers' professional development in a climate of educational reform. *Educational Evaluation and Policy Analysis, 15*(2), 129-151.

Los Angeles Unified School District. (2005). *Enrollment in APCS Courses, 2003-2006*. Los Angles, CA.

MacLeod, J. (1987). *Ain't no makin' it: Aspirations and attainment in a low-income neighborhood*. Boulder, CO: Westview Press.

Margolis, J., Holme, J. J., Estrella, R., Goode, J., Nao, K., & Stumme, S. (2003). The computer science pipeline in urban schools: Access to what? For whom? *IEEE Technology and Society, 22*(3), 12-19.

Oakes, J. (1985). *Keeping track: How schools structure inequality*. New Haven: Yale University Press.

Oakes, J. (1990). Opportunities, achievement, and choice: Women and minority students in science and mathematics. *Review of Research in Education, 16,* 153-222.

Oakes, J. (2002). Education inadequacy, inequality, and failed state policy: A synthesis of expert reports prepared for *Williams v. State of California*. [On-line]. Available: www.decentschools.org

Parks, C. (2003). The theory of everything. *American Association for the Advancement of Science Minority Scientists Network.* [On-line]. Available: http://nextwave.sciencemag.org/cgi/content/full/2003/11/20/4?template=msp

Schofield, J. W. (1995). *Computers and classroom culture*. Cambridge, England: Cambridge University Press.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher, 15*(2), 4-14.

Shulman, L. S. (2000). Teacher development: Roles of domain expertise and pedagogical knowledge. *Journal of Applied Developmental Psychology, 21*(1), 129-135.

Stephenson, C. (2005). Entering the forbidden forest of teacher certification. [On-line]. Available: http://blog.acm.org/archives/csta/2005/07/entering_the_fo.html

The Joint Task Force on Computing Curricula: IEEE Computing Society & Association for Computing Machinery. (2001). *Computing Curricula 2001: Computer Science*. [On-line]. Available: http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum*. New York, NY: Association for Computing Machinery, Computer Science Teachers Association.

Vegso, J. (2005a). CS bachelor's degree production grows in 2004; Poised for decline. *Computing Research News, 17*(2), 3.

Vegso, J. (2005b). Interest in CS as a major drops among incoming freshmen. *Computing Research News, 17*(3), 17.

Vegso, J. (2006). Drop in CS bachelor's degree production. *Computing Research News, 18*(2), 5.

Warschauer, M. (2000). Technology and school reform: A view from both sides of the track. *Educational Policy Analysis Archives, 8*(4).

Willis, P. E. (1981). *Learning to labour: How working class kids get working class jobs*. Aldershot, Hampshire: Gower.

Direct reprint requests to:

Dr. Joanna Goode
3735 Kincaid St.
Eugene, OR 97405
e-mail: goodej@uoregon.edu