-- PLEASE DO NOT CITE --

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

Lost in Translation: Gender and High School Computer Science

Prologue

Carolina is one of six females taking the Computer Programming course at East River High School, a predominantly Latino/a high school located in a low-income neighborhood on the east side of Los Angeles. She is one of the top students in her Honors Geometry class and has a strong history of academic achievement, particularly in mathematics and science. Carolina was recruited, along with other similarly strong female mathematics students, by a well-meaning and committed mathematics teacher, who had been assigned to teach Programming and was eager to increase female enrollment in what had been a predominantly male classroom. As the course evolved, the teacher, who was teaching himself programming while running the course, decided to take advantage of the more knowledgeable students by pairing "experts" with "novices." With two exceptions, these pairs included an older, more tech-savvy male—the "big brother"—and a younger, inexperienced female. Despite the girls' high standing in high level mathematics classes, the students quickly fell into "traditional" roles, with girls taking on more "secretarial" tasks. Carolina, who found this arrangement to be frustrating, states:

> *We [the girls] called them "big brothers"….I would type and he [partner] would like think about it—we would both think about it but he would like be the one with the most ideas…I was not always able to understand the ideas. Sometimes I just went along with it.*

Meanwhile, across town in a school on the more affluent west side of Los Angeles, Grace, a Pilipina senior at Canyon High, is the only female in her AP Computer Science course (APCS). Unlike the other female student who quit after the first semester, Grace decided to "stick it out" because, as she states, she is "not a quitter." Moreover, Grace is very interested in technology and plans to major in computer engineering. Grace became hooked onto computers because of her interest in cell phone technology. In the Philippines, there are many cottage industries *focused upon making the cell phone more fun, unique, and "hip" and this was fascinating to Grace, piquing her curiosity and prompting her to pursue computer science. In addition, Grace's father has experience working with computers, particularly in web design, which Grace believes also influenced her technology interests. Grace lives in downtown Los Angeles, but travels a long distance to Canyon High because of its reputation as a good school, with a full range of course offerings.*

Grace had a difficult time being the only girl in APCS, especially after her other female classmate dropped out after the first semester. Most of the males in the class were already quite knowledgeable about computing, having spent years of play time at home and with friends at the computer. In class they tended to dominate the discussions, and often diverted the teacher's attention during class time on debates about material that extended beyond the scope of the class. Grace's confidence in her ability to shine in this subject was shaken by her experience in this class:

> *Oh, gender is like a big thing, it's also like if you're like a girl, and if you go to a class that's mostly male dominant, you get like… you feel inferior because there's all these guys and you're the only girl in there, so it feels like….[like] they're gonna like be like better than you and sometimes you feel bad, like oh my gosh, all these guys, and what am I against them, you know? … So, I don't know. Sometimes I feel like they have like a better start than me.*

Introduction

For the last three years, we have been engaged in research that examines why so few African-American, Latino/a, and female students are learning computer science at the high school level. While we emphasize that race is a salient factor for students in the context of schools, this chapter focuses on the issue of gender, specifically the dearth of females in high school computer science classes. Considering how today's technological tools have become more mainstream, and women now rival their male counterparts in the use of communication technologies (Lenhart, Rainie, & Lewis, 2001; Goode, 2004), we were interested in understanding how and why the gender gap in who is learning how to invent and create new technologies stubbornly persists. In 2001-02, at the nation's Ph.D.-granting departments of computer science and engineering, just 7% of computer science bachelor's degrees -- the standard credential for obtaining positions in software design -- were awarded to African-Americans and Latino/as of both sexes, and just 18% to women (Taulbee Report Survey, 2002)

The trend is reflected in high school as well where only 17% of the APCS exam-takers are women and the combined percentage for African-American and Hispanic students hovers around 6% (College Entrance Examination Board, 2003).

While home is where many adolescent male students get introduced to computing, and where many hone their computing skills and knowledge through game playing, exploration and tinkering, school is an important site for being exposed to and learning computer science, especially for students who don't have the resources at home and/or peer and social networks outside of school that nurture along any budding interests. While multiple pathways into the world of computing certainly do exist (Jesse and Malcolm, this volume), enrollment in computer science programs and classes are an important indicator of how few women are learning and/or being engaged with this subject. Yet, as the opening vignettes convey, computer science in the schools often reflects the culture and domain of a sub-set of boys who are already immersed in the activity and knowledge of computing when entering the class (Schofield, 1995; AAUW, 2000; Margolis and Fisher, 2002) For girls who enroll in computer classes beyond the introductory level, their experiences are too often (though not exclusively) discouraging ones.

In this chapter, we present four themes that suggest some reasons why and how high school female students are --or are not-- drawn into the field of computer science through their high school experiences. These themes are derived from interviews with high school students and educators and our observations of their school contexts. First, despite the national and local initiatives to "bring schools into the 21st century," we have discovered how *few* computer science learning opportunities actually exist at the high school level, especially in schools that serve communities of color. Even at schools that are "heavily wired," computer science is too often interpreted as "computer literacy" and only low-level user skills are taught. And, despite the importance of computation across various disciplines and occupations, at the high school level the integration of computer science with other subjects in high school is hard to find. Thus, we found that many women of color simply do not have the *opportunity* to study computer science. Second, we found that notions of *relevance* play an important role in influencing females' choices to enroll or not enroll in computer science classes. A limited and narrow presentation of what computer science is and what computer scientists actually do impacts students' take on how computer science could further their academic and career endeavors. Third, for the female students who do take computer science, we have witnessed an accumulation of negative experiences in classroom settings, as captured by the opening vignettes, where greater male technology experience/expertise and female social isolation and insecurity are part of the cultural landscape. We have found that most of the females in our study lack the kinds of technology-oriented peer social networks enjoyed by the tech-savvy ("techie") males that we interviewed. Given that most of the male tech-savvy students reported that a majority of their computer science knowledge was acquired at home or with friends, the importance of play, experimentation, and social networks in computer science learning cannot be understated. Fourth, all of these experiences are then compounded by the way that computer science is motivated and "interpreted" for the students. Despite the critical thinking and problem-solving skills that are the foundation of computer science, we witnessed how the curriculum of most computer science classes at the high school level are missing a higher-order thinking focus. Unfortunately, educators routinely fail to provide a compelling context and motivation for learning computer science for students who may not be singularly interested in the computer. Moreover they often neglect to create learning environments that make the connections between computer science and the academic interests and pathways of "non-traditional" students, especially females. Our chapter concludes with an account of an action-research component of this study which strives to build a cadre of teachers as they work on redefining computer science for their students by restructuring the content and pedagogy in their classrooms to attract and retain increased numbers of female and other underrepresented students.

Theoretical Framework and Data Collection Strategies

Our framework for investigating the computing gender gap at the high school level focuses on the complex interaction of factors that impact students' "decisions" to study (or not study) computer science. It is a model formulated by Oakes (1990) for examining the low numbers of women and non-Asian minorities in mathematics and science education. This model takes into account the interaction of structural aspects of schools, such as tracking, with relevant psychological and sociological influences, such as gender and racial identity, peer dynamics, etc. We are guided by the theoretical proposition that there is an intricate interplay between students' interest in computer science in school and the structural and psychological constraints both at home and at school, and within and between social groups. For instance, as Oakes' (1985) research shows, and which our study affirms, student placement in classes often reflects the perceptions and biases of teachers and counselors, rather than the abilities or aspirations of the students. This is just one example why we must study both the structural as well as the psychological factors which promote the participation of girls in high school computer science classrooms.

We also consider what makes computer science "unique" and different from mathematics and science. The widespread representation of males as the computer science "doers" and experts is further exacerbated by the informal technology knowledge of boys and girls. For a sub-strata of boys, video games and hours of related tinkering and experimentation are the "hook" which gets them interested in computer science. These games, designed by males for boys, are not pulling girls in to the same extent. In addition, the field of computer science is distinguished from other mathematics and science courses by the fact that most educators themselves have little computer science experience and tend to be unfamiliar with the field in general. As a result, many educators are often not in a position to acquaint girls with and encourage them to pursue a course of study in computer science.

The heart of our research data is collected from observations and interviews with students and educators in three public high schools located in the Los Angeles Unified School District, one of the nation's most racially diverse school districts: a predominately Latino/a school (East River), a predominately African-American school (Windward), and a racially mixed high school (Canyon). We have conducted over 200 interviews with students and teachers. Our primary data collection strategies included weekly classroom observations, formal interviews and informal discussions with students, teachers, and administrators. We interviewed a minimum of sixty students from each school who were either engaged in a technology course or an advanced mathematics class. In addition, a total of four focus groups were held with students of color and female students at Canyon High School, whose purpose centered on the question of underrepresentation of both race and gender in computer science. We interviewed teachers who currently or formerly taught technology courses or advanced mathematics courses as well as key administrators at each school. Finally, we held regular meetings (once every three months) with four to six teachers at each school who either taught a computing course or who taught mathematics and worked with students of color. The foci of these meetings were threefold: 1) to receive feedback on our research, 2) to engage them in discussion about the state of computer science at their school, and 3) to better understand the how these teachers and administrators perceived their student body and how this perception influenced prioritization (or lack thereof) of specific curricula.

Extensive field notes were taken during interviews, observations, focus groups and discussions and were analyzed alongside the formal interviews. Interviews averaged about 40 minutes each and were transcribed and then coded and analyzed using Atlas ti software.
Our student interview protocol consisted of open-ended questions that covered the following domains:

- Academic background (educational history, academic interests, current class schedule, etc.)
- Computing history and affinity towards/current knowledge of computing (i.e. when/where they first learned how to use computers, what they currently know how to do on computers, what they like to do on computers, and what don't they like about computers?)
- Experience in computer classes
- Mathematics background and interest
- Computing at home (i.e. what kinds of hardware/software access did they have, what did they use the computer for at home, did their families use the computer and how knowledgeable were family members about computing?)
- Friends and computing (i.e. did they engage in computing activities with friends? Did they have friends who had extensive knowledge about computers? What were their friends' attitudes about the computer or about people who are interested in computers?)
- Perceptions of computer science and computer scientists

This article pulls from data from two (East River and Canyon) of the three schools we researched, because these were the two schools that offered computer courses beyond the introductory level, and this provided some semblance of a "computer science pipeline[1]" at the time of our research. It is important to note that we are not attempting to prove particular hypotheses, nor are we trying to empirically test a particular theory in order to arrive at a finding that can be generalizable to a particular populace. What we are trying to understand more in-depth are the dynamics in the schools we are studying so we can then generate theories that could help explain the gender gap in computer science at the high school level. We employ qualitative methods, using the words of our research subjects and our experience and insights as observers within their context, in order to create a more detailed picture of this phenomenon. It is

[1] In the context of our research of the educational opportunities at the high school level, we define the "computer science pipeline" as the curriculum and opportunities available to students to learn computer science within their schools.

our belief that by doing so, we will be better equipped to eventually devise interventions that honor the complexity of this issue.

School Settings

East River: "Technology Rich but Curriculum Poor"

East River High School[2] is located in a predominantly working-class community east of Los Angeles, in an area surrounded by industrial warehouses. Latinos/as make up 99% of the school population, many of whom are first and second generation immigrants. Four out of five students qualify for the free/reduced lunch school program, and 30% of students are English language learners. To accommodate 4,700 students, East River operates on a year-round three track calendar, so while 2/3 of the students are in school at any given time, 1/3 are on vacation (or "off track"). Ironically, while this school has been "red-flagged" for low test scores, it also has a history of being "technology-rich." A cadre of older, male teachers at this school has been the frontrunners in the school district in bringing computers into the classroom and wiring the school onto a network. In fact, at East River, the school's 4.8-1 student to computer ratio is better than the state average of 5.3-1. Moreover, one hundred and four classrooms are connected to the Internet (California Department of Education, 2003). The girls at East River excel in mathematics courses, and consistently outnumber their male peers in Advanced Placement mathematics courses (California Department of Education, 2003).

Despite the physical presence of a large quantity of networked computers, East River has never offered APCS, and only offered Programming on one track for two semesters. The computing curriculum at the school is largely vocational and not used for advanced academic study. Students who enroll in introductory computer courses learn keyboarding and word processing, while students in the popular Internet Publishing class follow a scripted curriculum to learn to make drawings, business cards, and to duplicate advertisements from the yellow pages. Students must take the Internet publishing course for four semesters before they are permitted to create web pages.

Rather than acquiring the knowledge that allows one to create and problem-solve with technology, the computing to which these students are exposed teaches only basic user skills. As one student stated, "From what I've heard it's like they make them type, and then they copy,

[2] Names of schools and students have been changed to protect the confidentiality of students and educators.

paste, copy, paste, copy, paste." Even though computers are equally distributed in each classroom, students do not report use of computers in their academic courses. Students interested in academic, challenging computing experiences at East River are denied an important learning opportunity. While many of the educators at East River view higher level computing classes as inappropriate due to the low levels of English language and basic skills, there are students who are interested in and would benefit from a rigorous computing curriculum. For example, Gabriela, for example, a self-identified "techie," notes that:

> *Well, there are a lot of people into computers, but they don't take the classes because they feel that it's beneath them. Well, it's pretty much true because half of the classes here are like basic typing and stuff, more than half of everyone knows how to do that anyway.*

Several years ago, a Programming class was offered on one of the school's tracks, but was taken off the curriculum when the school was flagged for low test scores. Because of low-enrollment, the principal felt he could not justify the class expense when the school had testing pressures to address. As we also noted over two years of observations, the content of the course itself consisted of mostly low-level syntax-oriented tasks, and the class was not a success for the few female participants, as reflected in the opening vignettes.

Canyon: A Segregated Pipeline

Unlike East River, which is surrounded by industrial warehouses, Canyon High School, located in the western part of the city, is encircled by multi-million dollar mansions and has an unfenced, green, tree-filled campus. The breezy sea-air is much gentler than the stuffy, smoggy air that East River students breathe. Despite its location in a White, wealthy section of Los Angeles, the school population is ethnically diverse. Canyon, a math-science magnet and charter school, currently takes students from over 100 different zip codes. In the 2002-2003 school year, 43% of its students were White (this figure includes a large number of students of Eastern European descent), 24% African-American, 24% Latino/a, and 8% Asian American/Pacific Islander. The school serves relatively few students who qualify for free or reduced-price lunch: just 24% are eligible (compared to 62% county wide). Furthermore, only 6% of the students are not yet proficient in English. Of the three schools in our study, the parents of Canyon students have the highest number of years of formal education: 81% have some college education (versus 55% countywide), and 63% have a college degree (compared with just 34% countywide).

Reflecting the community in which it is situated, Canyon is one of the more well-resourced schools in the school district. There are several computer labs on campus and every non-technology oriented classroom has at least four computers in it. Unlike East River, Canyon offers a complete computer science pipeline, including Data Processing (which has a programming component to it), Programming, and Advanced Placement Computer Science (APCS.) The instructor for all three of these courses does not yet hold a teaching credential, but has worked in the technology industry as a programmer and as an IT manager. The curriculum for the APCS course covers data structures, algorithms, and the management of complex, large scale programs. This college-level course incorporates a larger problem-solving orientation and is more aligned with the computer science field than any other class we observed. While the presence of more authentic computer science content seems to be in place here, the class enrollment reveals a very segregated pipeline.

Despite its racial and ethnic diversity, African-American, Latino/a, and female students at Canyon High are seriously underrepresented in the upper-level computer science class. While Data Processing and Programming classes, which satisfy a school technology requirement, have a fairly diverse population of students, the numbers of African-American and Latino/a, and female students dramatically drop away at the AP level. In the 2001-2002 school year, out of fifteen students, there were only two females (one Pilipina and one Persian, but the latter dropped out halfway through the year), one African-American male, two Latino males, and the remaining students were White males (several of whom were Russian immigrants). In the 2002-2003 school year, there were again fifteen students, but only three were female (two Asians and one White). That year, there were no African Americans and only two Latinos enrolled in the course.

Reduced Access to Computer Science

The disparities between East River and Canyon illustrate how computer science course offerings in this school district fall along racial and socio-economic lines. APCS (the benchmark course for higher-level computer science) is far more likely to be offered in schools serving more affluent White and Asian students (such as Canyon) than in schools serving large numbers of low-income students of color, schools that rarely offer courses beyond low-level vocational courses (such as East River) (Goode, 2004). We have found that a vocational, rather than an academic, model of computing dominates the computer science curriculum in schools serving predominately low-income students of color. This vocational model of computer science too often reflects the low expectations a school has for its students. Yet, despite the views of many educators at schools such as East River, that higher level computing classes would be inappropriate for their students due to the low levels of English language and basic skills, we encountered students who would be quite capable to enroll and learn computing beyond the introductory level.

When we began our research in 2001, all of the three high schools in our study offered computer programming and two of the three offered APCS. Yet, by the 2002-3 school year, the second year of our study, two of the three schools eliminated their computer science classes above the introductory basic skills level. The reasons offered for these course reductions are many: testing and the accountability movement that has relegated computer science to being an "extra," low expectations for students, lack of available teachers and training, and a system-wide misunderstanding of the interdisciplinary and academic worth of computer science. Indeed, in the midst of a national and local educational campaign to bring the schools "into the 21st century," computer science offerings in this district are being cut and reduced. We have also found that while the amount of technology in a school is often presented as an indicator of the quality of education, the presence of computers in the schools can be deceiving. Schools can be simultaneously "technology rich" and "curriculum poor." These access and curriculum issues then impact female students' opportunities to learn computer science, especially female students and students of color in general.

Though Canyon High and East River High differ drastically in terms of student demographics and educational resources, both schools consistently have a disproportionately low number of female students in computer science courses. In the following sections, we present several themes that address the interplay between school structure and psychological dynamics that we have found to impact female students' engagement with computer science.

Perceptions of Computer Scientists and the Field

When asked to describe their image of a computer scientist, interviewees at all three schools came up with inconsistent and uninformed descriptions, having never encountered a computer scientist and being unsure of what a computer scientist does. Several students cited Bill Gates as their primary image of a computer scientist. Others guessed that computer scientists "walk around in a white lab coat," merging their images of biological or chemical laboratory scientists with computer scientists. Almost every interviewee pictured in their minds a White or Asian male, some adding that they were "geeks."

For most of these students who have not experienced adults or other students doing interesting things with technology, their images of who works in computer science comes largely from popular culture. When asked how these images came to their minds, most students, male and female alike, cited the media as their primary source. As Nicole, a Pilipina Programming student at Canyon states:

> *N: I think that more guys are, like, into computers than are there girls `cause in a lot of pictures that I see about, like, computers and stuff, mostly the users are guys that I can see in pictures. There's not really as much women as there.*
> *I: And when you say "these pictures," do you mean, like, in magazines?*
> *N: Magazines, yeah. Magazines, newspapers, books, it's basically all guys.*

Print media are not the only arenas wherein the White or Asian male is portrayed as the quintessential prototype for a computer scientist. Students referred to the same stereotyped image in movies and in television as well. It is, as one student called it, the "Hollywood description." While this "Hollywood" depiction may not be entirely accurate, for students living in a media-saturated society who have no access to people in the field, the Hollywood image translates into a perceived reality. Moreover, the under-portrayal of women in computing is reaffirmed within educational settings. Though 72% of California's teaching force is women, only a quarter of the APCS teachers are women (Goode, 2004).

Breaking the Image of the "Lone Programmer"

For our female interviewees, what is perhaps more unappealing than the stereotyped or "Hollywood" image of a computer scientist is the notion that computer scientists are anti-social and that the nature of their work is solitary and isolating. Though practicing computer scientists work in groups to plan and implement solutions, many female students, when describing their image of computer science, speak of a "lone" programmer, staring at a computer. Grace, an APCS student who expressed a great deal of interest in technology, was turned off to programming because of this particular vision of computer science, stating:

> *I think I prefer the hardware side more because programming is just not my style because I don't like to sit at a computer all day and just stare at a screen.*

This image of computer science as solitary work, of men sitting in front of a computer, "working on it 24/7," as one female student stated, is not only a turn off, it is also misleading. Computer scientists at the university level and those working in the field indicate that a large portion of their job involves working as part of a team to achieve their given solutions. Yet the image of the solitary, geeky, overworked male computer scientist persists, not only because of what students see in the media but because this image of computer science work is often affirmed through high school computer science curriculum. For instance, at Canyon all programming assignments are individually-based projects. There are no group assignments. We have observed a learning atmosphere which perpetuates the image of lone programmers working on a formal process of writing a program. Despite the rich collaboration of designers across disciplines in the academic domain of computer science, not once did we see teachers encourage group work as a valued component of the design process. Nor were creative solutions or alternative approaches encouraged, a critical trait in the work of professional computer scientists. In other words, these classrooms furthered an implicit message that this is a solitary business.

Hollywood, print media, and the classroom all perpetuate erroneous images of computer science and computer scientists. For students with no access to people working in the field or who are technology enthusiasts, these arenas form the foundation of their understanding of what it may be like to work in this field, thereby helping to shape their decisions to study or not to study computer science. Social networks, therefore, play a key role in providing a better understanding of the realm of possibility provided by computer science and in piquing the interest of students who might not otherwise picture themselves working in this field or succeeding in a computer science class.

Importance of Social Networks

Family backgrounds and peer networks are very critical to consider when one examines who does or does not enroll in computer science. In APCS at Canyon, all of the most tech-savvy males we interviewed in the APCS class had prior programming experience that they received from home. Most of these students became interested in programming because of their love of video games and their desire not only to understand how these games were created but also to attempt to create games of their own. These students reported learning programming by tinkering with the computer, reading books on their own, and learning through friends and/or relatives who either worked in the field or had extensive programming knowledge. Not only did these techie students have social networks that helped them to further their knowledge, but they also had the financial resources to purchase all the latest software and hardware, a fast internet connection, etc. Several of the Canyon techie males owned more than one computer, and being able to build a computer from scratch seemed almost common knowledge. Two of the students we interviewed even set up a kind of business for themselves where they assembled or troubleshot computers for different clients. These male students' prior knowledge was so extensive that they claimed their primary motivation for taking the APCS class was to "have fun," as they felt the curriculum was too easy for them. In fact, they made no secret about the fact that they were unchallenged by the curriculum. On more than one occasion, these students were observed rolling their eyes during class discussions or making snide remarks about what they construed to be their female teacher's lack of knowledge. These actions not only reflected a disrespectful attitude, but they also created an intimidating environment for the female students who were not as experienced as their male techie counterparts.

At East River, networks for the most tech-savvy males also existed, though overall, students had far less access to technology at home compared to Canyon students. Still, like at Canyon, the males in programming at East River arrived at class with more gaming knowledge and other computing experiences at home as compared to the female students. This expertise impacted both the composition of the programming course and, consequently, the classroom culture. Fernando, a 12th grader who had enrolled in the course for two consecutive years, was considered by all of his classmates and his teacher to be the most knowledgeable programming student. He had been turned on to programming a few years earlier by a male friend, who encouraged him to build and host websites for extra money, an endeavor which served as a part-time job for him throughout high school. Fernando's own pursuit of more computing knowledge at East River inspired three of his closest friends to enroll during Fernando's second year in the programming course. And while we witnessed Fernando frequently and patiently aiding his posse of friends with code, as well as many of his neighboring male classmates, he persistently resisted helping the remaining students in class, particularly the four females, with their assignments. When asked about his least favorite aspect of the course, he confirmed that he found having to assist other students in class frustrating. Leo, Fernando's most tech-savvy friend, furthered this sense of animosity, asserting, "My least favorite [part of the class] is that there are people that ask a lot of dumb stuff, stuff they should know." This hostility did not escape the attention of the girls, as will be described later in this paper.

It is important to note that familial networks played an extremely important role in piquing girls' interest in computer programming. With the exception of one student, all of the females we interviewed at Canyon who actively chose to enroll in the Programming or APCS course because of interest in the subject matter[3] had a family member who either shared an interest or worked in the computer field. Two interviewees had fathers who worked in web design and these students took the course, at least in part, to understand their fathers a little better. One of these interviewees stated that she was curious about the kind of work her father did because she always saw him in front of the computer, occasionally yelling at it when he couldn't get something to work. Another interviewee worked part time for her father, who has a garage web business, and she wanted to learn more about what she could do with computers. Another female interviewee, whose father is an electrical engineer and whose brothers are also studying electrical engineering in college, took the APCS class because she was annoyed at being left out of technology-centered conversations with her father and male siblings.

While knowing someone close to them who was an expert in the computing field did not ensure that the females in our study pursued a sustained course of study in computer science, of importance is the fact that it helped pique their interest in the subject. Moreover, it appeared to make them more comfortable in acting upon these interests by enrolling in a computer science course. But the nature of the social networks for females and males were qualitatively different. The female students in our study were not part of a socially-based computing community, as were the boys who were gamers or enthusiasts who regularly attended computer shows. Nor did the females have the same range of technology knowledge as did the APCS males. Likewise, most of the East River students came from families who did not have parents at home with

[3] As opposed to taking it because of a graduation requirement or because it was the only class that fit into their schedules.

computing knowledge to impart to their children, and their computing resources at home were much more limited than in the Canyon homes. A question that emerges then is what should the schools be doing to compensate for these types of social networks and communities that are available for some students and not for others, yet appear to be so critical to students' jump start into computing?

Students without technology-oriented familial or social networks find themselves at a grave disadvantage when it comes to succeeding in a computer science class, for they are without the critical resource and reference point enjoyed by more privileged, tech-savvy students. Not only does the absence of these networks affect their ability to succeed in the computer science classroom, it also serves as an obstacle to truly understanding the possibilities inherent within the field of computer science. Moreover, it renders it difficult for students to see the connections between computer science and other courses of study and/or career paths.

Connection (or Lack Thereof) to Career Choices

At East River, many of the Latina students we interviewed were college-bound honors students from advanced mathematics classes. Among them were students who planned on being doctors, lawyers, architects, engineers, and teachers.[4] They were active and committed learners, who spent time between academic terms at special mathematics inter-sessions because of the scheduling interruptions of their year-round school. Despite being the most academically elite students in the school, these female students had little understanding of how computer science could support their academic and career plans. Teri, for example, strives to be an architect, but has never heard of computer-aided-drafting. Kara aims to pursue criminology as a career but cannot imagine how computing might aid her profession. Most of these students argue that their core academic subjects are their priorities for getting into college and pursuing their career. They value "creativity" and purpose but unfortunately do not associate these qualities with what turns out to be their very limited understanding computer science.

[4] The career choices that these top achieving female students identify with present an interesting topic for further exploration. We refer readers to the research of Marcia Linn and Janet Hyde (1989) who concluded that a major sex difference in interests in mathematics and science is perceived usefulness, and the research of Eccles (1994) that found that "women select the occupations that best fits their hierarchy of occupationally relevant values" and that helping other and doing something worthwhile for society is high in that hierarchy (Eccles, 1994, p.600). Further discussion on women's values and career interests are found in Margolis and Fisher, (2002, p.55).

The story is similar for the female students at Canyon. Examples of some of the career choices of our female interviewees include doctor, lawyer, teacher, PR professional, and writer. As with the female students at East River, most of the female students we interviewed at Canyon did not make a connection between computer science and their career choices. However there were several female students who reported finding programming interesting ,who were intrigued by its creative aspects, and who felt an intense sense of accomplishment when solving a difficult programming task. Yet these positive attributes proved not to be strong enough to persuade the majority of the female students we interviewed to continue to pursue computer science in high school. Of the nineteen Canyon female interviewees, only two who took a lower level course in the CS pipeline said that they would pursue further study. These two students mentioned their love of problem solving and logical reasoning as a key impetus for continuing to pursue computer science, describing it as "challenging" and "fun."

We were intrigued, then, as to why so few female students who expressed interest in computer science continued with the course of study at Canyon, a school with the resources that would allow for the continuance of this CS learning opportunity. Indeed, even at East River, despite the existing computer science vacuum, there were several female interviewees who also found programming interesting but expressed no desire to pursue the subject. For interested female students at both schools, the primary reason articulated for not pursuing computer science as a course of study was that it wasn't worth taking more advanced computer science classes because it didn't "buy" them enough academically. In other words, the decision not to pursue computer science was a strategic decision on the part of these students, a decision that they felt would preserve their competitive standing in the academic marketplace.

Competitiveness for College (Academic Relevance)

In today's increasingly competitive market for college acceptance, it seems almost naïve to assume that students choose their high school courses primarily because of interest in a particular subject of study. For students "in the know," i.e. high achieving students who are planning to apply at elite colleges and who understand how to work through the proper academic pathways, courses are treated as commerce. For instance, with the average weighted GPA of students accepted into UCLA or UC Berkeley resting at 4.24, college bound students are faced with intense pressure to take courses that will push their GPA higher in order to make them more competitive in their college applications. Indeed, many of the female students we interviewed were high achievers, focused upon succeeding academically and getting accepted into a good college. These students have very tight schedules, filled with classes that counselors tell them will look favorable in the eyes of college admissions counselors. Neither Programming nor APCS are marketed as one of those classes. The high achieving, college-bound females we interviewed claimed that they simply do not have the time in their schedules to pursue computing. Indeed, Programming at East River satisfies the same requirement as completing another technical art course, including floristry. At Canyon, not only is Programming *not* seen as a class that would pique the interest of college admission counselors, but it is also on equal level with Data Processing in terms of fulfilling their high school graduation requirement. Thus, students who are wary of risking their grade point averages often take the easier of the two courses.

University admissions guidelines play a confusing role here, too, and it is in this arena where the structure of high school clashes with state policy. While California state policy emphasizes the importance of technology in improving education, and millions of dollars have been spent creating digital high[5] schools and emphasizing computers as central to education reform, little has been done at the University level to affirm the importance of computing to education. For example, the University of California has established a list of courses that students are required to take to meet the minimum UC college eligibility requirements (also known as the "A-G requirements".) Yet despite the increasing importance placed upon technology by the state of California, the only computing course that satisfies any of the A-G requirements is APCS, a course not offered to students at East River.

Many elite students at East River, especially girls, express that they feel knowing computers beyond typing buys them little in terms of the immediate goal of being college eligible. And in a sense, they are right. At Canyon, where there is an abundance of Advanced Placement (AP) classes from which to choose, students are taught by counselors that, while all

[5] "Digital High Schools" were authorized by the 1997 California State Assembly bill for the Digital High School Education Technology Grant Program. The legislation specifies: "It is the intent of the Legislature that all high schools in the state become 'digital high schools' by the end of the first year of the 21st century and that these schools fully integrate computers, networks, training, and software to achieve computer literacy in all pupils and faculty and to improve academic achievement." Among the program objectives are "to provide all high school pupils with basic computer skills, improve pupil achievement in all academic subjects, and increase collaboration among high schools, post-secondary institutions, industry and community organizations."

AP courses increase your GPA, not all AP courses are considered equal. According to teachers and administrators we interviewed, students are told by their guidance counselors that there are prime AP courses that they should target, such as AP European History, AP English, and AP Government. Again, the notion of courses as commerce is helpful here. While taking an AP class will inevitably buy you something (in this case, an extra point for your GPA,) each course is treated as a commodity with differential value. This value, however symbolic (and erroneous), holds weight in the eyes of students in pursuit of higher education and influences decisions not to take courses that will in any way place their GPA at risk. In this sense, many girls' decisions to not enroll in computer science reflect a strategic decision to protect their college-going course schedule.

Yet, while it may not buy them enough in terms of college credits, the fact remains that females, however few, did take APCS at Canyon and Programming courses at both schools (when it was offered). Our data reveals that their experiences within this classroom environment tended to have a negative impact, not only in their affinity to computer science but also in their understanding of it as a subject of study.

Climate in CS Classrooms

Summary of observation field notes of a Canyon APCS class:

> *On this particular day, when the teacher wrote a problem on the board for the class to work through, three to four of the most tech-savvy male students shouted out possible methods for solving the problem, argued with each other, then got up from their seats*[6] *and stood in front of the board to discuss the problem, blocking the view from the rest of the class. This classroom discussion then turned into a long private debate amongst these male students. The teacher, who appeared pleased that the problem engendered so much interest and debate, did not do anything to include the other students. These silent students sat at their desks, listening to these students argue while looking at their backs. While the blocking of the whiteboard was not necessarily a common occurrence, the reduction of classroom discussion into a private debate amongst the class techies has been a common event in our observations. Also common was the teacher's lack of acknowledgment of this problem. Again, this appeared to stem not from a purposefully gendered agenda, but rather from the teacher's excitement that her students were engaging in debate, an excitement that kept her seeing and changing the dynamics of this debate in terms of who was participating and who was silent.*

[6] The APCS classroom setup is such that 6 computing tables are put together to form a hexagon, with 6 of these hexagonal formations spread out in the room. The techies sit together, covering the two hexagons closest to the teacher.

Several of the Canyon female interviewees commented on this particular dynamic, telling us that they felt intimidated by the large disparity in experience between themselves and the males in their computer classroom. Jennifer, an APCS student, noted how asking for help became difficult for her:

> *What I don't like? Well, I do think that it's a bit intimidating to be in the class when everybody else really thinks they have a strong handle on it and. Yeah, I don't know. It's all these other guys, they seem to know exactly what they're doing, and it's sometimes been difficult even asking for help.*

Elizabeth, another APCS student, described the male behavior as “showing off,” adding that it seemed to take up a great deal of class time:

> *Yeah, I saw some showing off. And yeah, it was like mostly, like, three or four people doing the talking. And like, they go off on other stuff about technology that we don’t really know about, or that I don’t even know about, like more advanced code or, like, I don’t know, like electronics and stuff.*

Grace described to us her difficult time being “the only girl in there,” especially after her other female classmate dropped out after the first semester. Our observations, coupled with our interviews with her, reveal that her feeling of conspicuousness and inferiority is not just an internal psychological response at being different, but that it is, at least in part, triggered by the beliefs and behavior of the males in the APCS classroom. One male techie scoffed at the difficulties experienced by not only the one female in the class, but by the Latinos as well. Privately, in an interview, this student ridiculed Grace’s and the Latinos’ lack of programming knowledge, calling their class work “a joke.” Moreover, this student stated that he and some of the other techie males in the class often made fun of these students, expressing irritation at what they felt was a curriculum that was watered-down to accommodate those with less experience.

We also witnessed more subtle but disturbing gender dynamics in the second round of observations in a East River Programming class. This programming class included a cohort of twelve older, more knowledgeable boys, and four younger, less knowledgeable girls. Unlike the previous year, however, the four females in this course were not honors students and did not know each other, but all had expressed interest in wanting to learn about computers. They also began the academic year in the course along with their male classmates. Their stories, however, reveal a more hostile programming environment than the previous year. The boys, mostly seniors with prior computing knowledge obtained at home, flaunted their knowledge, upsetting and agitating the girls.

Alicia told us:

> *The seniors make us feel bad…cause they think they're good. They are good. They'll be talking down to us…*

Cara says*:*

> *I don't like [the boys in the class]. Because they think they're all that cause they're seniors and they think we're the dumb freshmen. -- Cara*

This group of girls huddled around a couple of computers and worked together as a group, while the boys worked independently around them. The girls depended much more on the teacher for assistance, while the boys would ask a male neighbor for aid first, relying on the teacher as a last resort. Closer observation of the boys, however, showed that many of them "shared code" with each other in order to complete assignments. Often a male would allow a peer to copy his code, and then another, and so on. The girls were largely kept out of this "in the know" circle, and as a result, completed their assignments long after their male classmates, perpetuating their standing as the least knowledgeable students in the class.

Our observations of this class also exposed disturbing trends in the ways that the teacher treated the male and female students. For instance, the teacher would select one of the girls each day to return the attendance sheets to the office, a five minute trip, but in dozens of our observations, never once sent one of the male students. When a few computers were broken, it was the girls who were relieved of the assignment and allowed to work on homework from other courses in the classroom space of the lab.

Though the dynamics of this programming class were different from the "big brothers" class described in the opening vignette and the Canyon APCS class, the outcomes were largely the same with males occupying the more knowledgeable position and females always being in the less knowledgeable position.

Engagement in the Computer Science Classroom

The situation we observed in Canyon, the one school with advanced level computer science, is that the students' who enrolled in this class were predominately the tech-savvy males and the "converted." While APCS is formally designed as a first course introduction to the science of computer science, and the College Board (2002) suggests only intermediate algebra as a prerequisite, students who succeed in computer science often enter the course with a great deal of knowledge and experience in computing, having learned most of their computing from home and with peers.

Unfortunately, in neither school did we find a curriculum that successfully incorporated the interests of a broader segment of the student population, including women students, to the study of computer science. For instance, class assignments in the East River Programming class rarely played to the academic strengths or interests of the girls, and generally failed to capture the interests of the girls. It is important to remember that the girls enrolled in the class were honors mathematics students, who were top achievers in the school. One programming assignment, for example, required students to write a program that, when given dimensions of a several rectangular rooms in a house, would compute the cost of carpeting a house, a problem which at best uses middle school mathematics skills. This "problem" could easily be solved using simple multiplication and addition with a calculator, and failed to capitalize on the high-level processing power that the field of CS draws upon. Other assignments hovered at the same low-level implementation of programming and focused on basic trivia games; including a hangman game, a state capital trivia game, tic-tac-toe game, and a program which provides a test for students studying the area of basic geometric figures. Despite the fact that many of the females in the class excelled in their honors mathematics courses, their problem-solving skills and creative skills were rarely utilized in the programming class. In fact, none of these assignments legitimately represented the types of complex, real-life problems which would otherwise be difficult or lengthy to tackle without the field of computer science. Instead, students only encountered simple programs which focused on basic input and output processes. Students were never encouraged to define their own problems, or specify the parameters, an important component of programming. Nor were connections to other academic subjects (besides low-level mathematics) tapped into through the curriculum of this class.

At Canyon, the curriculum in the Programming class is vastly different from that of the APCS class. In the APCS class, much of the curriculum is geared toward honing problem-solving skills in order to best prepare students for the AP test at the end of the year. A typical lesson plan involves the teacher putting a sample AP question on the board, which the class is expected to discuss and try to solve together, after which time the teacher hands out a set of questions that the students are expected to answer individually. There are multiple avenues through which students can arrive at the answer. Their objective is to use their problem solving and logical reasoning skills as well as their knowledge of JAVA or C++ to come up with a solution. In contrast, the curriculum for the Introduction to Programming course is textbook-based. Students read assigned portions of the textbook and then try to re-create the project discussed in the given chapter. Sample projects include writing a program that calculates the timing for a traffic signal, a program where you can create movie tickets, and a program that calculates grade point averages. The textbook gives step by step guidelines on how to create a program for the given project, even offering the exact code needed. So in order to create the program, students need only follow the given sample in the textbook. Unlike in the APCS class, students in the Programming course simply follow directions as dictated in the textbook, using Visual Basic as their programming language. All programming students therefore follow a similar course in order to achieve the same solution for the presented problem. Thus, the course is closer to a cut and paste format than one that engages in problem solving and logical reasoning. This format is not only less challenging and creative for students, but it also keeps them from truly understanding the programming language itself and the multiple ways in which it can be used to solve a problem. This has been a critique given about the class by both male and female students alike.

And again, all assignments in the programming course are individually-based, with no group work or whole-class discussions. Students without knowledge of how computer science is used in the real world are therefore left with the impression that computer science is solitary in nature. This proved to be a turn-off for the females in our study who conceptualized computer scientists as “anti-social” and “isolated.” In short, the Programming course at Canyon reinforced negative stereotypes about computer science and computer scientists. In fact, like East River’s programming course, Canyon’s programming class did not provide students with a concrete idea of how computer science is applied in the real world, nor what the field of computer science currently looks like, rendering it difficult for someone with no prior knowledge of the field to find it appealing. Unfortunately, the Introduction to Programming course often failed to engage the interests of students and therefore served more as a turn off than as a stepping stone to APCS.

What we found was that computer science is too often de-contextualized and robbed of connections to subjects and arenas that interest the women in our study. This is similar to the findings in the AAUW 2000 report that found that "computer science instruction that emphasizes the "web" of associations between programming, design, and other areas of the curriculum would help to attract a more diverse group of learners, and would advance computer fluency for all students" (p.43).

Needed: CS Teachers to Re-vision an Alternative Classroom Culture and Pedagogy

In our research we have encountered countless well-intentioned educators who do not have access to the knowledge and resources required to present a more accurate and relevant computer science curriculum to students. We place our observations within the context of the immense challenges presented to computer science teachers: teacher education programs do not offer methods classes for computer science teachers, creating no clear pathway for becoming a computer science teacher. In fact, the computer science teachers we have encountered came from a variety of disciplines, including mathematics, English, social studies, music, art history, and business. As a result, few of these teachers themselves have a familiarity with the modern field of computer science. Also, unlike other teachers, computer science teachers rarely have a home department, resulting in limited (if at all) opportunities to collaborate with colleagues to develop curriculum and support their teaching endeavors. Additionally, these teachers have technical requirements they must work around, taking on an additional role as troubleshooter for the computers in the classroom. The constantly changing field of computer science also presents barriers to teachers who strive to keep up with the field. The programming language for APCS, for example, has changed from Pascal to C++ to JAVA, all within the last six years. Keeping up to date with these changes, without any professional development support, seems to be an insurmountable challenge. With such bare essential needs not addressed, it is not surprising that few teachers have time or will to think about alternative pedagogy and curriculum that can be meaningful for females and a more diverse pool of male students. And, yet, teachers are one of the critical gatekeepers assuring the existence and success of a diverse classroom, and this must be their challenge.

With a better understanding of the challenges encountered by high school computer science teachers, we felt compelled to utilize our resources and connections to work with a group of urban educators as part of an ongoing action-research relationship. This type of alliance combines theory and practice, places researchers alongside practitioners, and aims to create a reflective and iterative cycle where practice and theory inform each other to make change rooted in praxis (Avison, D., et. al., 1999). To this end, we have formed an educational partnership, the Computer Science Equity Collaborative, consisting of the UCLA Graduate School of Education and Information Studies, the UCLA School of Engineering and Applied Science, and officials from the Los Angeles Unified School District. The first endeavor of the collaborative was a 2004 summer institute with 25 local high school APCS teachers involved in a week of activities at UCLA, including discussions around issues of equity in computer science, computer science instruction, showcases of interdisciplinary computer science applications around campus, and the modeling of an engaging, active pedagogy, aimed at dispelling the image of computer science as an individualistic field about computers. Highlighting the multidisciplinary, problem-solving nature of computer science, we have challenged these teachers to take more active roles in the recruitment and retention of females and students of color in their classes. Moreover, we have garnered the commitment of district officials and school principals to support these teachers. Our hope is to keep this community of educators connected together to support each other, and we intend to learn what works in their classrooms and what doesn't. Rather than sitting back and passively researching classroom dynamics, we will engage in a partnership with these educators, trying to support their efforts to recruit and retain more females and other underrepresented students into their classes. Since our institute, the number of APCS classes doubled in the district, and female enrollment in APCS courses in LAUSD has increased 75%.

Summary

The challenge before educators is not just to recruit females and other underrepresented students into computer science and APCS courses, but to engage them and to provide them with support. Without meaningful assignments that immerse students in higher-order thinking tasks, and interdisciplinary contexts, many students will fail to see the relevance of computing in their own interests. As in a literature class, pedagogy for computer science class should not be reduced to a study of grammar and syntax. Often missing from advanced CS classes is the creation of a CS context that would capture the interests of a broader range of students, beyond (but not excluding) the technology-focused students. This requires consciously aware and committed educators who re-set the tone, hold a larger understanding of the computer science field, and challenge old notions of who is capable, able, and potentially interested in learning this subject. Taking into account students' motivation for learning, and what is meaningful for different groups of students, is critical. An important task will be to dispel the prevalent myths about computer science: that it is only about the love of the computer, that it is solitary, that it is focused on tinkering and gaming rather than being relevant to "real world" issues, and that only anti-social people engage in it.

We have witnessed female high school students who want to be more deeply involved with technology, who know it is part of their future, who dream big, but who have no understanding of what deeper involvement entails, or what preparation is required. This is a reflection of unequal home resources and lack of learning opportunities in the schools, and a "better start" for male students. The most pressing problem in schools serving the lowest income students is one of access---the courses simply aren't there. However, there are also the issues of teacher training and pedagogy—the need for teachers to be properly prepared and proactive in recruiting females and underrepresented students into their classes, and to alter their pedagogy to engage what has become the "non-traditional" student. We found that the scientific heart of computer science is "lost in translation" at the high school level, and as a result the field continues to lose the participation and interest of a broad layer of students, especially females.

REFERENCES

American Association of University Women. (2000). *Tech-Savvy: Educating girls in the new computer age*. Washington, D.C.: AAUW.

Avision, D., Lau, F., Myers, M., Nielson, P. (1999). Action Research. *Communications of the ACM, 42*(1), 94-97.

Burbules, N., & Callister, T. (2000). *Watch IT: The risks and promises of information technologies for education*. Boulder, CO: Westview Press.

College Entrance Examination Board. (2003). *Advanced Placement program California and national summary reports*. New York, NY: The College Board.

College Entrance Examination Board. (2002). *Computer science course description*. New York, NY: The College Board.

Eccles, J. (1994). Understanding women's educational and occupational choices. *Psychology of Women Quarterly* (18): 585-609.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy* (1st ed.). New York: Palgrave Macmillan.

Goode, J. (2004). *Mind the Gap: The digital dimension of college access.* Unpublished Dissertation, UCLA, Los Angeles, CA.

Lenhart, A., Rainie, L., & Lewis, O. (2001). *Teenage life online: The rise of instant-message generation and the Internet's impact on friendships and family relationships*. Washington, D.C.: Pew Internet & American Life Project.

Linn, M.D., and Hyde, J.S. (1989). Gender, mathematics, and science. *Educational Researcher* 18 (8):17-27.

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.

Margolis, J., & Fischer, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT Press.

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.

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

Cambridge University Press.

Taulbee Report Survey. (2002). *2001-2002 CRA Taulbee Survey*. Retrieved January, 2004, from www.cra.org/statistics/home.html.