# Book Reviews

DOMINIC W. MASSARO, EDITOR
University of California, Santa Cruz

## Gender and the Digital Divide

**Gender and Computers: Understanding the Digital Divide**
By Joel Cooper and Kimberlee D. Weaver. Hillsdale, NJ: Erlbaum, 2003. xi + 168 pp. Cloth, $19.95.

What influence does gender have on the digital divide? In preschool, both girls and boys show similar interest in computers (Williams & Ogletree, 1992). Studies show that by the beginning of middle school, boys' interest increases and girls' interest decreases (Anderson, 1993; Holis, 1985; Fetler, 1985; Chappel, 1997). The percentage of female full professors at computer science PhD-granting departments during the academic year of 1993–94 is a dismal 5.7% (Camp, 1997). What is more interesting is that while the percentage of women has increased in most professions that were formerly considered male oriented, the percentage of women graduating with bachelor's degrees in computer science has decreased from 35.8% in 1984 to 28.4% in 1996 (Robb, 2003).

Some say that there are at least as many women as men using the Internet (Losh, 2003). Is that not enough to show equity? Why are the percentages of computer science graduates or full professors important? The reason is that a computer science degree can put a person in the position of influencing the direction of what computer technology is created and how that technology is used.

A recent book by Joel Cooper and Kimberlee D. Weaver, two social psychologists, presents research that attempts to explain the gender gap in technology. *Gender and Computers: Understanding the Digital Divide* targets social scientists, educators, and parents with precollege children. The authors hope the book can be used by social scientists to identify additional avenues of research, educators to change classroom policy, and parents to provide equal access for both daughters and sons to computers in their homes. These are three important audiences, but the authors only partially succeed in speaking to all three and meeting their objectives to describe, trace causes, and offer suggestions for change.

In a useful categorization, Cooper and Weaver dedicate a chapter to each of the four areas of psychological factors they determined "underlie girls' and women's failure to realize their full computing potential" (p. 7). These areas are computer anxiety, effects of social context, expectations and the self-fulfilling prophecy, and stereotype threat. The chapter on computer anxiety makes the case that girls get anxious if they work with software designed for boys. If students are anxious, they avoid computers, and performance suffers. The chapter begins with a story about one girl's anxiety in response to being asked to play video games and effectively uses case studies to illustrate the implications of research findings for educational settings. The most promising implications for change are that computer attitudes are influenced by early experiences with computers and gender role orientations rather than biological sex. This chapter provides a useful summary of research, some conducted by Cooper and his colleagues working with elementary school children and university students.

In the next chapter, the authors argue that the social context has a strong influence on the attitudes and behaviors that lead to involvement with computers. The chapter begins with a review of classic social psychological research on how the presence of others affects individual actions and performance. They say "audiences increase performance on tasks on which people feel comfortable and decrease performance on tasks on which people feel anxious or uncomfortable" (p. 44). The authors extend this research to make the claim that girls' performance on computers will suffer when "the social context for technological learning includes groups of children" (p. 43). Research cited to substantiate this hypothesis looked at many settings in which children's performance on computers was affected by an audience. Included in the research cited to support the authors' hypothesis are studies using groups of pairs of children, each working at their own computer; groups in which each child has his or her own computer; and groups in which one child was at a computer, and one or more other children not working on computers were present in the room. The studies showed that solitary work (without an audience) resulted in the best performance for the girl, with girls outperforming boys in some private settings.

In the chapter on expectations and the self-fulfilling prophecy, Cooper and Weaver cite a great deal of research that does not focus on computer use. They start with the classic 1968 sociology of education study by Rosenthal and Jacobson that showed that children randomly designated as high achievers performed better than those their teachers believed were low achievers (p. 69). Cooper himself led other research with college students that showed how "(racial) stereotypes can lead to expectations. These expectations, in turn, lead to differential behavior by the 'perceivers' which can actually lead to expectancy-consistent behavior by the targets" (p. 74). Research by Cooper and others is cited to explain how students' own expectations about their performance can affect their relationship with the computer. Results indicate that girls attribute success on computers to luck and failure to lack of ability more often than boys and that boys attribute success to ability and failure to luck more often than girls. The authors further conclude that expectations about a generic (i.e., male) person result in software that is really designed for boys.

The next chapter applies research on stereotype threat to explain gender differences in computing. They explain, "The mere existence and knowledge of a negative stereotype causes anxiety and pressure in members of the stereotyped group. If women know that they are considered inferior in their math, science, or computer abilities, they are confronted with a predicament. They know that observers (e.g., teachers, employers, males) possess this stereotype and will use it as a lens through which their performance will be judged" (p. 95). The research suggests that stereotypes reduce performance among those who identify with the domain. So the more a woman identifies with being female, the worse she will perform in regard to computing if presented with the stereotype that males are better than females. However, this conclusion is based on extrapolations from other research; no research is cited that examines the effects of stereotype threat on women's attitudes or performance in computing.

Two final chapters offer suggestions for helping to bridge the gender digital divide. The first chapter focuses on what parents and teachers can do, such as raise their expectations about what their daughters and female students can do regarding computers. The second chapter offers suggestions that include structuring classrooms in ways to make the technology more appealing to girls and having single-sex schools.

### Weaknesses

The authors refer to many studies, including many that were conducted by Cooper with various colleagues, that trace causes for the gender differences in the use of computers or that form the basis for the authors' opinions regarding computers and gender. These are presented in clear language that will appeal to nonresearchers. However, Cooper and Weaver do not provide sufficient details for most of these studies so that researchers can weigh the importance of the studies' results. By aiming to appeal to a wide audience of parents, teachers, college educators, and researchers, the authors have limited the scientific usefulness of the book.

A related concern is the misuse of computer terminology. Throughout most of the second and fourth chapters of the book, the authors refer to computer-assisted instruction (CAI) programs as synonymous with information technology. This is not accurate. Here is a direct quote from the book: "Computer programs designed to help teach children their mathematics, language, science and social studies, known as *information technology (IT) programs* or *computer assisted instruction (CAI),* comprise a multi-billion-dollar industry" (p. 16). To make it clear that we have not misinterpreted the authors' intention, what follows is another quote from this same chapter: "They did not enjoy the eye–hand coordination (e.g., the pointing of a gun at an oncoming tank before a mathematics question could be answered) that is part and parcel of many IT experiences. To the contrary, girls preferred their IT programs to be more like learning tools" (p. 16). It would have been appropriate for the authors to replace "IT" or "IT programs" with "CAI" or "CAI software" throughout chapters 2 and 4. IT is a broader field, not just restricted to this one aspect of IT known as educational software or CAI. IT software includes business applications such as payroll and accounts payables and receivables, office applications such as word processors and database system software, and general-purpose software such as operating systems and CAI software. The authors also use the phrase "computer-assisted software" (p. 42) instead of the accurate "computer-assisted instructional software." In the preface, Cooper and Weaver suggest that those "who implement technology policy in the classroom will find important lessons here." They also state that "those who teach courses in human–computer interaction, human factors, . . . will profit from this book" (p. xi). Again, the authors' efforts to make the research accessible to a lay audience are admirable but undermine the legitimacy of the book with a more technical and scientific audience.

Another weakness is the unquestioning acceptance that there are gender-specific programming styles. Regarding university computer science programs, the authors state that the "work shows that men and women tend to prefer to program in different styles, and that the style that is taught in the vast majority of classes is that which is preferred by men" (p. 31). The authors' belief is based on research done by Turkle and Papert (1992), who described two styles of programming: "soft" (bottom up) and "hard" (top down). In the "soft" style, people construct their programs from smaller building blocks that they themselves construct. These building blocks are then used to construct larger building blocks until the entire program is complete. The "soft" style can be contrasted to the "hard" style in which standard, preprogrammed black box building blocks are used to construct programs. These black boxes are used opaquely, with little concern about the internal details. When using this "hard" style, the programmer must plan, in a well-structured way, the design of the final program. Cooper and Weaver did not include any reference to the optimistic belief stated in Turkle and Papert's 1992 article that Apple's Macintosh operating system and object-oriented programming methods hold promise for encouraging a "soft" style of programming.

Because of developments in computer languages, specifically object-oriented ones such as C++ and Java, we have seen university computer science programs move away from the pervasive style of the 1980s, where top-down methods were taught, to one in which problems are approached in both a bottom-up or "soft" and a top-down or "hard" fashion. Object-oriented methods are found in all phases of computer software development including requirement specification, design, programming, and testing. A simple Web search of just the introductory computer programming courses at Carnegie Mellon, Stanford, and the University of Illinois at Urbana–Champaign shows that all of these courses include object-oriented concepts. I would have expected Cooper and Weaver to have determined whether there were advances in computer science education and practice that accommodate object-oriented topics. If Turkle and Papert were right that men and women prefer to program using different styles, then given that they also stated that object-oriented programming "legitimates alternative methods" of programming (Turkle & Papert, 1992, p. 31), Cooper and Weaver's suggestion that multiple perspectives should be used when educating people to program is already being followed. Again, a careful editorial reading by a computer scientist would have identified this problem. Statistically significant results of research done to show that the use of object-oriented languages is preferred by women have not yet been collected. We view this as an area for new research given that Parnas has argued that computer scientists do not really know how to rationally, systematically, design software in a top-down way. He states,

> The picture of the software designer deriving his design in a rational, error free way from a statement of requirements is quite unrealistic. No system has ever been developed in that way, and probably none ever will. Even the small program developments shown in textbooks and papers are unreal. They have been revised and polished until the author has shown us what he wishes he had done, not what actually did happen. (Parnas & Clements, 1986, pp. 251–252)

A carefully constructed study of 41 first-year programming students, most with artistic backgrounds, found no significant difference between women and men in their attitudes regarding "soft" and "hard" styles of programming (McKenna, 2001).

As one of the recommended solutions for bridging the digital divide, the authors encourage teachers to search for gender-neutral software. It would have been useful to include examples of resources to help teachers and parents to find this gender-neutral software. These include software such as *The Zoombini's Logical Journey* or *KidPics,* two excellent examples of gender-neutral software that have been in existence for several years. (Hancock & Osterweil, 1996; Kafai & Soloway, 1994). In addition, Web sites such as "Through the Glass Wall" (http://www.terc.edu/mathequity/gw/html/gwhome.html) provide a useful analysis of these two pieces of software and other gender-appropriate mathematical software.

## Recent work

A final critique of the book focuses on the authors' claim that women probably will suffer in a public computing context (p. 43). This claim can be challenged by new research on how both male and female college students benefit from working truly collaboratively. The authors include only one study of true cooperative learning, and its results were that women benefited from the cooperative setting (p. 50).

This new research challenges the assumption that children should work alone at the computer. The solitary approach is used at all levels of schools, despite research that suggests wide benefits of cooperative and collaborative work (Azmitia, 1996). A study by the American Association of University Women suggests that this may be one reason why women stay away from computer-related fields of study. These reasons include

The perception that a career in computer science is not well rounded or conducive to family life

The belief that work in the field is done in a competitive rather than a cooperative way

The perception that work in the field is done in a solitary way

Concerns about safety because much solitary after-hours and weekend work is required in computer laboratories

Because introductory computer programming is a gateway course for anyone considering a career in IT, recent research at the university level looked at the social aspects of introductory computer programming courses to see whether a change in the social structures could effect a change in the gender composition of the field. Rather than enforcing the solitary programming paradigm, pair programming was used. Pair programming was developed as part of a computer software development method called extreme programming by Kent Beck and his colleagues, Ron Jeffries and Ward Cunningham, and puts two people together at a single computer station (Beck, 2000). One person acts as the "driver," working the keyboard and mouse. The other person acts as the "navigator," guiding his or her partner by actively choosing the best methods for coding the software while scanning for problems in the code. The roles are reversed after a period of time, and any code created by only one of the partners is reviewed by the collaboration before inclusion. The research involved more than 500 students and suggests that pair programming holds promise for increasing women's participation in IT. Performance on independently taken exams was not affected by pair programming; those programming alone performed as well as those working in pairs. Additionally, the retention of women and men in computer science and other related university majors significantly increased. Many results of this research were particularly dramatic for women (McDowell, Werner, Bullock, & Fernald, 2003). Because of such dramatic results at the university level, we are experimenting with pair programming at the middle school level (Werner et al., 2004).

These recent results challenge the authors' hypothesis. The authors use the research on social facilitation as the basis for their hypothesis that girls would perform better with computers when working in a solitary environment, whereas boys work better with computers when others are present. This research looks at performance quality and the presence of an audience. Cooper and Weaver discount other research that "girls tend to do quite well in so-called 'cooperative learning tasks'" where the outcome of a lesson depends on the cooperation and communication between peers (p. 44). Cooper and Weaver cite research that concludes that boys perform better with an audience, especially when the task is something that boys do well. The authors state that boys are already receiving more time on computers, CAI use in the elementary schools focuses on topics that are of interest to boys, and teachers favor boys over girls. These are given as the basis for saying that boys perform better with computers than girls do. To this is added the fact that boys like an audience, and the authors conclude that boys are set up to continue to perform better with computers than girls perform with computers because classroom education means there always is an audience. We would like to argue that classroom education does not mean there always is an audience. We do not have performances with audiences here; instead, we can design true collaboration into our educational settings, something that traditionally has helped girls and women succeed and enjoy their educational tasks.

Linda L. Werner
*UCSC Baskin School of Engineering E2-249*
*1156 High Street MS SOE3*
*Santa Cruz, CA 95064*
E-mail: Linda@cse.ucsc.edu

Jill Denner
*Senior Research Associate*
*ETR Associates (Education, Training, Research)*
*4 Carbonero Way*
*Scotts Valley, CA 95066*
E-mail: Jilld@etr.org

## References

Anderson, R. (1993). *Computers in American schools 1992: An overview.* Minneapolis: University of Minnesota Press.

Azmitia, M. (1996). Peer interactive minds: Developmental, theoretical, and methodological issues. In P. B. Baltes & U. M. Staudinger (Eds.), *Interactive minds: Life-span perspectives on the social foundation of cognition* (pp. 133–162). New York: Cambridge University Press.

Beck, K. (2000). *Extreme programming explained: Embrace change.* Reading, MA: Addison-Wesley.

Camp, T. (1997). The incredible shrinking pipeline. *Communications of the ACM, 40*(10), 103–110.

Chappel, K. (1997). Investigating the impact of elements of educational mathematics software on girls' attitudes. *Journal of Educational Computing Research, 17,* 119–133.

Fetler, M. (1985). Sex differences on the California Statewide Assessment of Computer Literacy. *Sex Roles, 13*(3–4), 181–191.

Hancock, C., & Osterweil, S. (1996). Zoombinis and the art of mathematical play. *Hands On! 19*(1). Cambridge, MA: TERC. Retrieved September 7, 2004, from http://www.terc.edu/handson/s96/zoom.html

Holis, B. (1985). Sex related differences in attitudes toward computers: Implications for counselors. *School Counselor, 33,* 120–130.

Kafai, Y., & Soloway, E. (1994). Computational gifts for the Barney generation. *Communications of the ACM, 37*(9), 19–22.

Losh, S. (2003). Gender and educational digital gaps: 1983–2000. *IT & Society, 1*(5), 56–71. Retrieved August 26, 2004, from http://www.ITandSociety.org

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. *Proceedings of the 25th International Conference on Software Engineering (ICSE'03),* 602–607.

McKenna, P. (2001). Programmers: Concrete women and abstract men? *Journal of Computer Assisted Learning, 17,* 386–395.

Parnas, D., & Clements, P. (1986). A rational design process: How and why to fake it. *Transactions on Software Engineering, SE-12*(2), 251–257.

Robb, D. (2003). IT gender gap widens. *Datamation,* January 6. Retrieved August 24, 2004, from http://itmanagement.earthweb.com/career/article.php/1564501

Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior, 11*(1), 3–33.

Werner, L., Denner, J., & Bean, S. (2004). Pair programming strategies for middle school girls. *Proceedings of the 7th International Conference on Computers and Advanced Technologies in Education* (pp. 161–166). Calgary, AB, Canada: IASTED.

Williams, S., & Ogletree, S. (1992). Preschool children's computer interest and competence: Effects of sex and gender role. *Early Childhood Research Quarterly, 7,* 135–143.

## A New Beginning for Empirical Dream Research

### The Scientific Study of Dreams: Neural Networks, Cognitive Development, and Content Analysis

By G. William Domhoff. Washington, DC: American Psychological Association, 2002. 209 pp. Cloth, $49.95.

William Domhoff's *The Scientific Study of Dreams* breaks new ground in the field, not by proposing grandiose, premature answers to questions about the nature of dreams but rather by showing us with unprecedented clarity and scope where we have erred in the past and allowing us to start over again. In a careful, reasoned critique, Domhoff demonstrates the shortcomings of the dominant dream theories