To appear in *Computer*, IEEE Computer Society Press, November 2006.

# Gender HCI: What About the Software?

**Laura Beckwith[*], Margaret Burnett[*], Susan Wiedenbeck[†], Valentina Grigoreanu[*]**

[*]Oregon State University and [†]Drexel University

[*]{beckwith, burnett, grigorev}@eecs.orst.edu, [†]Susan.Wiedenbeck@cis.drexel.edu

## Abstract

*Although there have been gender studies designed to understand and ameliorate the low representation of females in computing, there has been little emphasis on how software itself affects females' and males' performance in computing tasks. Building upon theories and research on gender differences from a number of domains, we have been investigating how gender differences interact with software.*

## Introduction

In high school, "Ashley's" career plan was to become a graphic designer. However, when attempting to tackle Flash programming in the graphics design course, frustration set in. Although Flash is used by graphic designers, the original WYSIWYG programming style has largely given way to a Java-like language aimed at software developers who use Flash. Then the need to do web programming arose as well. Ashley decided that learning Flash and web programming was too great a barrier, and instead majored in art.

What were the real causes of Ashley's difficulties? Possibly Ashley's problem-solving style, learning style, or level of confidence made learning these software tools seem more formidable than it would to someone else. Gender differences in these and other domains, such as psychology, marketing, and neuroscience, strongly suggest that females process information and problem solve in very different ways than males do (c.f. [2, 14]). We have been investigating whether these sorts of gender differences should be taken into account in the design of software.

The possibility that *software* could erect barriers to females has only recently emerged [2, 12]. There is, however, research on other relationships between gender and computers, such as on gender differences with computer display hardware [21], on gender-oriented marketing strategies for web-based shopping [23], and on computer game software content [8, 10]. There has especially been work on factors affecting females' interest in computer science as a career choice, including academic "climate", educational strategies, human resource management, and social/cultural factors [9, 16, 18, 24, 25]—just about everything *except* the way the software works.

Gender HCI is the term we use to refer to research into how software relates to gender differences. Our particular focus is on how "gender-neutral" software works, not on gender-oriented content. Specifically, we have concentrated on software aimed at supporting everyday users doing problem solving. Examples of this sort of

software include spreadsheets, CAD systems, macro builders, educational software authoring systems, and media authoring systems.

There are two important reasons to care about issues in supporting both genders' use of problem-solving software. First, such experiences could impact the "pipeline" of women in information technology, essentially closing off the pipeline's beginning. As with Ashley, if a female's early experience with software that is supposed to support her problem-solving efforts is negative and discouraging, how likely is she to decide to eventually choose a career in information technology? The second reason is the productivity of end-user problem solvers, regardless of whether they might someday become software developers. Without taking relevant gender differences into account in the design of problem-solving software, barriers can be introduced into the software that interfere with the success of half the population the software is intended to support.

## Method

Our method for conducting this investigation consists of four steps: (1) draw from theory and previous gender difference empirical work from other domains—such as computer confidence, perceived risk, information processing, computing gaming, and technology adoption models—to hypothesize gender issues and their causes that could arise from gender-based differences in the use of problem-solving software, (2) use empirical methods to investigate whether these issues do actually arise in problem-solving software, (3) use the results of the first two steps along with qualitative empirical work involving low-cost prototyping to derive and refine approaches to address the issues, and (4) use quantitative empirical methods to evaluate the effectiveness of the approaches.

## What We've Learned So Far: Four Experiments

### Experiments #1 and #2: Gender Differences in Feature Acceptance

Several of the hypotheses we developed in the first step of the above method predict that in a software environment with problem-solving features, gender differences will have a significant impact on adoption and usage of these features, due at least in part to differences related to risk perception and confidence.

Gender differences regarding computer confidence have been widely studied, revealing that females (both computer science majors and end users) have lower confidence than males in their computer-related abilities [2, 7, 15]. Of particular pertinence is the concept of *self-efficacy* [1]. Self-efficacy is a person's judgment about his or her ability to carry out a specific course of action to achieve a goal. According to self-efficacy theory, high self-efficacy is critical in problem solving because self-efficacy influences the use of cognitive strategies, the amount of effort put forth, the coping strategies adopted in the face of obstacles, and the final performance outcome. This, combined with other research finding females are more risk-averse than males, implies that females' low self-efficacy regarding computing tasks will render them less willing to explore and adopt new features.

We began our investigation into the possibility of gender and its impact on software-based problem-solving activities in Experiment #1, a qualitative reanalysis of feature usage from data collected in a prior experiment that

was not related to gender [19]. We found that differences in feature activity showed a striking tendency to align by gender. For example, Figure 1 shows two activity profiles of one male and one female user that were fairly representative of other experiment participants of their respective genders. (Our participants are all non-computer science students, primarily recruited from the business school.) As the figure suggests, the males' amounts and types of feature usage were overall quite different from those of the females.

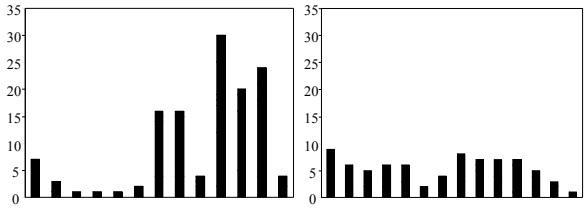



**Figure 1. Activity profiles of one male (left) and one female (right) in a problem-solving software environment.**

Following up on Experiment #1, Experiment #2 was a quantitative laboratory study, in which 27 male and 24 female end users debugged spreadsheet formulas as their problem-solving task [3]. The setting was a research spreadsheet system containing a variety of features that have previously been shown to help end users test and debug spreadsheet formulas [6]. One of the spreadsheets and the testing/debugging features are shown in Figure 2. The features were partitioned into three categories for analysis: *familiar* features (such as editing formula textboxes), features *taught* during the experiment's tutorial (checkmarks and arrows), and unfamiliar features that were *untaught* (Xs). We wanted to find out how much males and females used these three types of features, the relationships between their

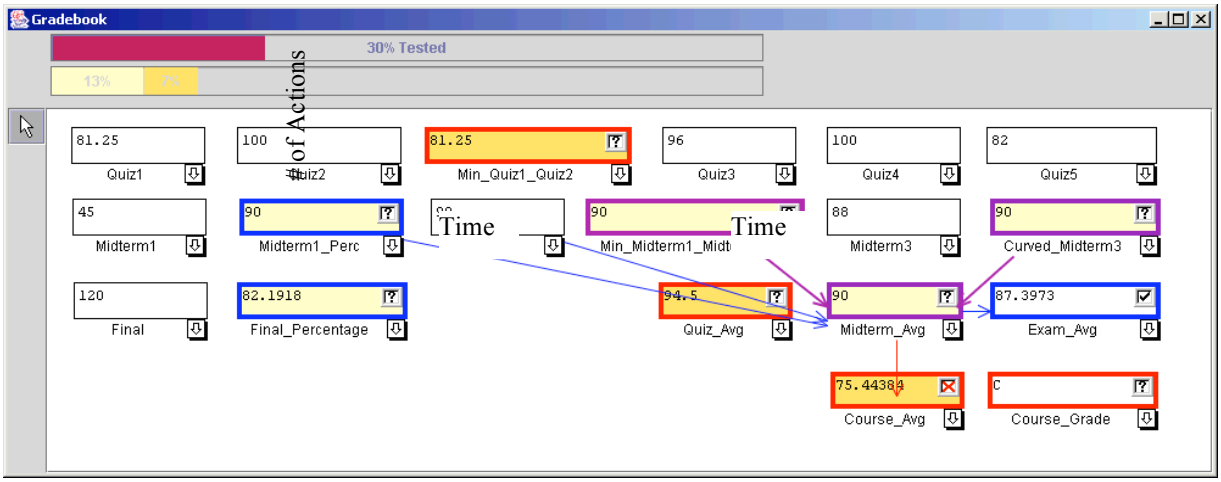



**Figure 2. Three problem-solving features in the research spreadsheet system [6]: (1) When users notice a correct value, they can "check it off" (cell `Exam_Avg`). Untested cell borders are red; they turn more blue as they become more tested according to a dataflow-based test adequacy criterion. These same colors in the (2) arrows show dataflow relationships among the cells and how tested those relationships are. (3) If instead a user notices an incorrect value (as in cell `Course_Avg`—the value is obviously too low) they can mark it with an X. As a result of this X and the checkmark in `Exam_Avg`, eight cells are highlighted as being possible sources of the incorrect value, with some deemed more likely than others. There are also tool-tip based explanations available for the features (not shown).**

| Result Type | Outcome |
|---|---|
| Self-efficacy | Mann Whitney: ***p*<.018** (Males higher) |
| Self-efficacy predicts effective feature usage | Linear regression (Males no, females yes):<br>Males: $p<.551$, $F(1,25)=.365$, $R^2=.015$<br>Females: ***p*<.046**, $F(1,22)=4.52$, $R^2=.177$ |
| Likelihood of initially trying new features:<br>Type Taught<br>Type Untaught | ANOVA (Males used earlier):<br>Taught: ***p*<.005**, $F(1,49)=8.69$<br>Untaught: $p<.073$, $F(1,40)=3.40$ |
| Likelihood of adopting new features for repeated use:<br>Type Taught<br>Type Untaught | Various methods (Males more):<br>ANOVA: ***p*<.03**, $F(1,49)=4.971$<br>Fisher's Exact Test: ***p*<.01** |
| Bugs fixed | Mann Whitney: $p<.651$ (No difference) |
| Bugs introduced | Fishers Exact Test: ***p*<.015** (Females more) |
| Thought features would take too long to learn | Mann Whitney: ***p*<.017** (Females more) |

**Table 1. Statistical results of Experiment #2. Bold values are statistically significant at p<.05.**

feature usage and their self-efficacy, and the implications on task success.

Table 1 shows several statistical results from this experiment. There were indeed gender differences in self-efficacy, with females significantly lower. Further, females were significantly slower to try out the new features and also were significantly less likely to adopt them for repeated use. For example, Figure 3 depicts the mean number of minutes into the task before participants first tried out features in each category. As the figure shows, females gravitated toward the familiar feature of editing formulas, whereas males were more likely to try less familiar features early.

Interestingly, the relationship between self-efficacy and feature usage was different for males than for females. For females, low self-efficacy was predictive of low effective usage of features (Figure 4). For the males, however, self-efficacy was not a predictor.

An obvious question then arises: were the females simply right about their low belief in their own abilities? The evidence does not suggest this; rather, it points toward females' low self-efficacy and their low usage of the new problem-solving features as hindrances to their performance. First, there was no significant difference in the males' and females' performance in fixing the bugs provided. However, females were significantly more likely to introduce new bugs that were never fixed. This seems tied to their heavy reliance on formula editing instead of the other problem-solving features: formula editing is, of course, the only way new bugs can be introduced. Second, females agreed significantly more than the males with this statement: "...I was afraid I would take too long to learn [untaught feature]". However, despite the gender differences in *expectation* of their ability to learn the untaught feature, there were no significant gender differences in *actual* learning of the feature. This seems to be a case of *inappropriately* low self-efficacy of the females inhibiting their use of a new feature, which in turn prevented them from receiving its benefits. In effect, their inaccurate assessment of ability became a self-fulfilling prophecy.

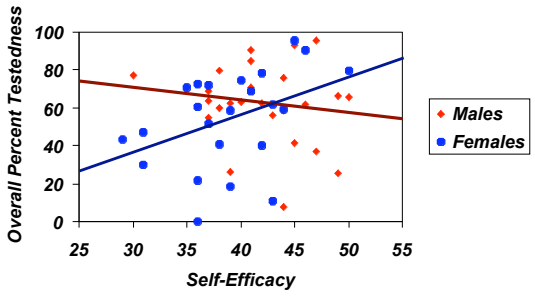



**Figure 4. Females' self-efficacy was a significant predictor of their effective use of the "check-off" feature, as the positively sloping line shows. For the males, however, this was not the case; note how most of the males clustered in the upper half, regardless of their self-efficacy.**

### Experiment #3: Attitudes toward New Features

We have conducted several other experiments, both qualitative and quantitative. In all of them, some form of gender difference in self-efficacy is present. The qualitative experiments consistently illustrate a common attitude in the females in which low self-efficacy seems to hinder their ability to cope with new, unfamiliar features.

For example, Experiment #3 was a qualitative think-aloud study that added a "guards" feature to the research environment, with red circles akin to those of Excel's "data validation" feature which circles out-of-range values. Female participant F1 exemplifies the attitude that has consistently emerged from low self-efficacy females in these experiments:

> F1: "What's this little arrow doing? They're everywhere! So, I need to take this—oh, my goodness. Now what's happening? … too much happening."

Experiment #3 also revealed an interesting difference in the ways features were perceived by males and females. For example, female F3, in using the new guards feature, said:

> F3: "I don't think that you can get a -5 on the homework. No, it can't be. So 0 to 100 [is the guard I'm entering], ok. Ok, hmm… So, it doesn't like the -5 [...]. They can get a 0, which gets rid of the angry

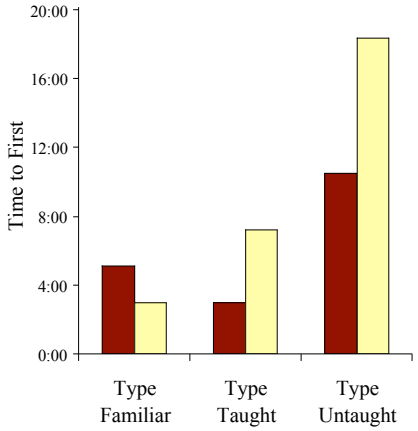



**Figure 3. Males (dark bars) first used new (taught and untaught) features significantly earlier than females (light bars).**

red circle."

In contrast to F3's focus on the guard feature as a way to get her spreadsheet to work correctly, the following male's initial focus was on the feature itself:

> M4: "The first thing I'm going to do is go through and check the guards for everything, just to make sure none of the entered values are above or below any of the ranges specified. So, homework 1—actually, I'm going to put guards on everything because I feel like it. I don't even know if this is really necessary, but it's fun."

Despite his initial interest in the feature for the fun of it, the male soon transitioned to its problem-solving advantages to the task at hand:

> M4 (continuing): "...It looks like the guard on the sum of the first two homeworks is wrong, isn't it? Is this even necessary, should I even be doing this? Alright, what are you doing now?" [*A red circle appeared because his guard did not agree with the computer generated guard.*] "Ok, so it doesn't like my guard apparently. Ok, ah ha! The reason I couldn't get the guard for the sum to be correct is because the sum formula is wrong."

In fact, the above gender differences in views toward the same feature are consistent with reports of gender differences regarding motivation for using technology, for majoring in computer science, and how children talk about the use of technology [5, 11, 15]. In particular, the male participant's use of the guards "because I feel like it" is similar to oft-reported reasons males give for majoring in computer science: technology for the fun of it.

### Experiment #4: How the Tinkering Factor Fits In

Such playful experimentation as male M4's, or tinkering, is often encouraged in educational settings because of its expected educational benefits [20]. It is possible that tinkering in problem-solving software may yield similar positive benefits, if tinkering encourages users to incidentally and iteratively gain knowledge of the software features, as M4 did. But education literature reports that tinkering is a strategy more often adopted by males than females [13, 17, 22], possibly leaving females without the advantages that can be gained from this strategy.

An exploratory analysis of Experiment #2's data showed that males did significantly more tinkering with the untaught feature than females did, which may explain the males' greater interest in this feature. Given the strongly suggestive evidence from Experiments #2 and #3 of tinkering differences by gender, we embarked on Experiment #4 to investigate whether the self-efficacy factor would discourage females from tinkering, and whether this would be tied to negative outcomes for them. Experiment #4's design [4] was similar to Experiment #2's but added a comparison with a second software environment, which added features designed to provide greater support (see Table 2). We term the second software environment *high-support*, and the original software environment *low-cost*. For both genders, we measured participants' tinkering, self-efficacy, and effectiveness in the high-support and low-cost environments.

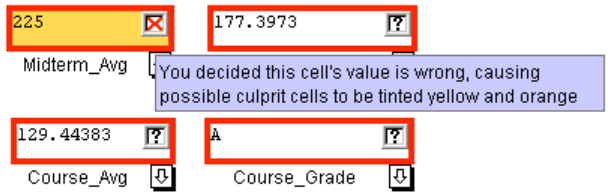
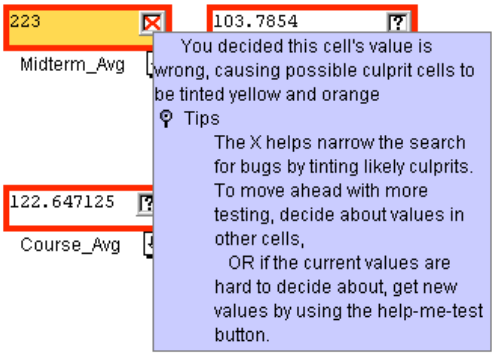
| Environment | Example |
|---|---|
| Low-Cost Environment: as in Experiment #2. This environment was expected to be more encouraging of tinkering. |  |
| High-Support Environment: included additional explanations, tentative ("maybe") check-offs and X-outs, and a "help me test" scaffolding feature. These support features were expected to bolster low self-efficacy participants, but the additional richness of the features also added to the cost and complexity, requiring extra clicks to access the additional support features and producing more feedback for the users to interpret. |  |

**Table 2. In Experiment #4, participants were assigned to one of two environments.**

The results revealed several differences between the genders in their tinkering practices and how these related to self-efficacy and to effectiveness. Males tended to be comfortable with tinkering. It seems to have been a match to many of their problem-solving styles. But this was not wholly a good thing for the males: males also had a tendency to tinker to excess.

Figure 5 depicts these tinkering tendencies and their relationships to effective outcomes, and Table 3 shows some of the statistical results. On the female side of Figure 5 (left), starting at bottom, the rectangle depicts amount of tinkering. Regardless of software environment, increased tinkering was significantly predictive of females' testing effectiveness, as the red arrow pointing up shows. Following the next red arrow up shows that, just as in the previous study, increasing testing effectiveness was significantly predictive of increased bugs fixed for the females.

Moving across to the male side of Figure 5 (right), the predictive relationship between effective testing and bugs fixed was also true for the males, but unlike the females, the tie between males' tinkering and testing effectiveness was not significant. Further, males' tinkering was *inversely* predictive of bugs fixed.

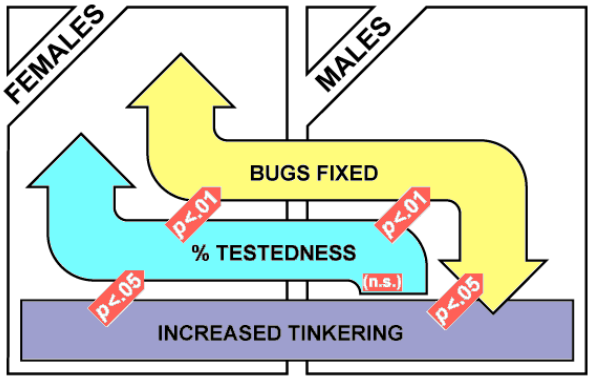



**Figure 5: Males' and females' tinkering affected their debugging effectiveness in unexpectedly opposite ways. Direction of stylized arrows depicts increase/decrease in a measure, and red arrows show significance of the regression relationships between measures.**

| Result Type | Outcome |
|---|---|
| Tinkering. | Unpaired t-test $p<.054$, t= -1.96, df = 74 (Males more) |
| Tinkering as a predictor of effective feature usage (measured as final percent testedness) in testing | Linear regression (Males: predicts *less* effectiveness, females: *more* effectiveness):<br>Males: ***p*<.05**, F(1,34)=8.04, $R^2$=.19<br>Females: ***p*<.05**, F(1,38)=4.63, $R^2$=.11 |
| Effective feature usage in testing predicts effective bug fixing | Linear regression (Males yes, females yes):<br>Males: ***p*<.01**, F(1,34)=27.16, $R^2$=.44<br>Females: ***p*<.01**, F(1,38)=10.51, $R^2$=.22 |
| Pausing:<br>Pauses<br>Predictor of effective feature usage in testing<br>Predictor of understanding the features | <br>ANOVA: ***p*<.05**, F(1,74)=4.22 (Females higher)<br>Linear regression: ***p*<.01**, F(1,74)=31.3, $R^2$=.30<br>Linear regression: ***p*<.01**, F(1,74)=14.11, $R^2$=.16 |
| Tinkering differences by environment | ANOVA 2(gender) x 2 (environment): (Environment affected the genders differently)<br>Main effect of environment: ***p*<.01**, F(1,72)=7.15<br>Interaction effect: ***p*<.05**, F(1,72)=4.42 |
| Tinkering predictive of post-task self-efficacy change:<br><br>Intermittent tinkering in high-support environment<br>Tinkering in low-cost environment | Linear regression:<br>Males: no predictive effects.<br>Females *decrease*: ***p*<.05**, F(1,21)=6.32, $R^2$=.23<br>Females *increase*: $p<.10$, F(1,15)=3.51, $R^2$=.19 |

**Table 3. Statistical results of Experiment #4.**

These differences between males' and females' tinkering behaviors and the benefits that came from these behaviors seem tied to two factors. First, some males exhibited a tendency to tinker without pausing much. In fact, males paused significantly less often than the females, and when they did such "pauseless" tinkering, their effectiveness suffered. Research from the education field has shown that if students are given "wait-time" of three seconds or more after a classroom response, their critical thinking about that response improves [20], and we found similar results: pauses after any action were predictive of more understanding and more effective use of problem-solving features.

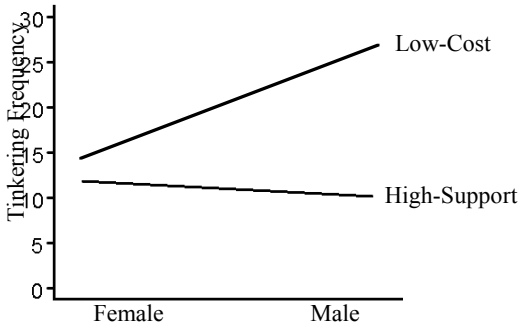



**Figure 6. This interaction plot depicts the gender x environment interaction in tinkering frequency.**

Second, the differences in the software environments affected male and female tinkering behavior differently. As Figure 6 points out, females tinkered about the same amount in the two software environments, but males' tinkering, which was about the same as the females' in the high-support environment, skyrocketed in the low-cost environment. In the low-cost environment, where tinkering was easy to do, males were willing and eager to tinker to excess, which was not the case for females.

The tinkering results suggest that tinkering (in either environment) helped females gain valuable information about the features, leading to positive outcomes. However, there were both positive and negative ties between tinkering and females' self-efficacy. As with our other experiments, pre-task self-efficacy was an important predictor of effectively using the features. However, one type of tinkering in the high-support environment, a pattern of intermittently tinkering in fits and starts, was predictive of a drop in post-task self-efficacy. The more the females tinkered intermittently in this environment, the larger their drop in self-efficacy.

One possible interpretation, in light of the findings from Experiments #2 and #3, is that, as the females received increased feedback from the high-support environment through their tinkering, they became ever more convinced that they would not be able to master its more complex features. In the other, less complex environment, there was suggestive evidence linking increased tinkering frequency to increased self-efficacy for the females after the task. This suggests that females' self-efficacy may be more sensitive than males' to tinkering with software whose user interface seems to contain obstacles, such as by being overly complex. The males did not experience changes in self-efficacy tied to tinkering in either environment.

In summary then, the males' biggest problem related to tinkering was that too much tinkering interfered with their effectiveness. The females' biggest problem was that, although in some circumstances tinkering helped with their self-efficacy, under other circumstances it interfered with their self-efficacy.

## Conclusion

There is ample evidence that gender differences exist in the ways people solve problems. Our results show that these differences are highly relevant to users' ability to gain benefits from the features that exist in software. As we continue this work, we hope to ultimately identify specific features and attributes of software that present barriers, so that alternative choices can be made available to users of the software.

We offer the following advice to usability professionals. First, gender differences often are not revealed simply through feature usage counts. Rather, gender differences are manifested subtly, in users' early exploration of features, in adoption patterns for long-term use, and in style of usage. Second, for male users, watch for excessive tinkering. Although males' willingness to tinker can be an asset, excessive tinkering can be harmful to males' task success. Environment factors that make tinkering too readily available may encourage tinkering to excess. Third, for female users, watch for ineffective usage of or avoidance of particular features. Signs of this in females, particularly low self-efficacy female users, may indicate a need for more supportive feature design.

It is important to an organization to encourage a diversity of ideas contributing to their efforts, in order to maintain their creative edge. This is true for the employees who *create* the software an organization builds, and it is also true for the employees who *use* software in problem-solving activities—like Ashley.

In fact, "Ashley" is a male. His story is true. Ashley went on to a college career in art, and ultimately won the most prestigious academic award his university bestows. He enjoys art, but regrets his decision not to pursue graphic design. Now that he has graduated, the barriers to making the switch back to graphic design are even higher, because he no longer has access to the educational support structures available to students. Still, at home after work, he is working to overcome the barriers that prevent information technology from being a good fit to his strengths.

Although gender differences in self-efficacy, motivations, problem-solving styles, learning styles, and information processing styles are all implicated in this issue, it is important to remember that no single female is likely to have every trait statistically associated with females, nor is any single male likely to have every trait statistically associated with males. For example, some males process information in the comprehensive style statistically associated with females, and some females process information in the more linear style associated with males. Thus, designing software in ways that support these differences does not penalize either gender—it helps everyone.

## Acknowledgments

We thank Alan Blackwell, Thippaya Chintakovid, Curtis Cook, Xiaoli Fern, Michelle Hastings, Sienna Hiebert, Derek Inman, Cory Kissinger, Joseph Lawrance, Vaishnavi Narayanan, Kyle Rector, Shraddha Sorte, and Sherry Yang for their significant contributions to this research. Special additional thanks to Alan Blackwell regarding Figure 5. This work was supported in part by Microsoft Research, by NSF grant CNS-0420533, by an IBM Faculty Award, and by the EUSES Consortium via NSF grants ITR-0325273 and CCR-0324844.

## References

[1] Bandura, A. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review 8*(2), 1977, 191-215.

[2] Beckwith, L. and Burnett, M. Gender: An important factor in problem-solving software? *IEEE Symposium on Visual Languages and Human-Centric Computing Languages and Environments*, September 2004, 107-114.

[3] Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S., and Hastings, M. Effectiveness of end-user debugging features: Are there gender issues? *ACM Conference on Human Factors in Computing Systems*, April 2005.

[4] Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A. and Cook, C. Tinkering and gender in end-user programmers' debugging. *ACM Conference on Human Factors in Computing Systems*, April 2006.

[5] Brunner, C., Bennett, D., and Honey, M., Girl games and technological desire, In J. Cassell and H. (Eds.), *From Barbie to Mortal Kombat: Gender and Computer Games*, Cambridge, MA: MIT Press, 1998, 72-88.

[6] Burnett, M., Cook, C., Rothermel, G. End-user software engineering. *Communications of the ACM*, September 2004, 53-58.

[7] Busch, T., Gender differences in self-efficacy and attitudes toward computers, *J. Educational Computing Research* 12, 1995, 147-158.

[8] Cassell, J., and Jenkins, H. (Eds.) *From Barbie to Mortal Kombat: Gender and Computer Games*, MIT Press, Cambridge, MA, 1998.

[9] Cohoon, M. Toward improving female retention in the computer science major. *Communications of the ACM* 44(5), May 2001, 108-114.

[10] Gorriz, C. and Medina, C. Engaging girls with computers through software games. *Communications of the ACM* 43(1), January 2000, 42-49.

[11] Hou, W., Kaur, M., Komlodi, A., Lutters, W. G., Boot, L., Cotton, S. R., Morrell, C., Ozok, A. A., & Tufekci, Z. (2006). "Girls don't waste time": Pre-adolescent attitudes toward ICT, *ACM Conf. Human Factors in Computing Systems*, Montréal, Canada, April 2006, 875-880.

[12] Huff, C., Gender, software design, and occupational equity, *SIGCSE Bulletin 34*(2), 2002, 112-115.

[13] Jones, M. G., Brader-Araje, L., Carboni, L. W., Carter, G., Rua, M. J., Banilower, E. and Hatch, H. Tool time: Gender and students' use of tools, control, and authority. *Journal of Research in Science Teaching 37*, 8 (2000), 760-783.

[14] Kucian, K., Loenneker, T., Dietrich, T., Martin, E., and von Aster, M., Gender differences in brain activation patterns during mental rotation and number related cognitive tasks. *Psychology Science* 47(1), 2005, 112–131.

[15] Margolis, J., Fisher, A., Miller, F., Caring about connections: Gender and computing, *IEEE Technology and Society Magazine 18(*4), 1999, 13-20.

[16] Margolis, J. and Fisher, A., *Unlocking the Clubhouse: Women in Computing*, MIT Press, Cambridge, Massachusetts, 2002.

[17] Martinson, A. M., Playing with technology: Designing gender sensitive games to close the gender gap. Working Paper SLISWP-03-05, School of Library and Information Science, Indiana University, http://www-slis.lib.indiana.edu/research/working_papers/files/SLISWP-03-05.pdf, accessed September 12, 2005.

[18] Othman, M. and Latih, R. Women in computer science: No shortage here! *Communications of the ACM* 49(3), March 2006, 111-114.

[19] Robertson, T. J., Prabhakararao, S., Burnett, M., Cook, C., Ruthruff, J., Beckwith, L., and Phalgune, A. Impact of interruption style on end-user debugging, *ACM Conference on Human Factors in Computing Systems*, Vienna, Austria, April 2004, 287-294.

[20] Rowe, M. D. *Teaching Science as Continuous Inquiry: A Basic (2nd ed.).* McGraw-Hill, New York, NY 1978.

[21] Tan, D., Czerwinski, M., and Robertson, G., Women go with the (optical) flow, *ACM Conf. Human Factors in Computing Systems,* April 2003, 209-215.

[22] Van Den Heuvel-Panheizen, M. Girls' and boys' problems: Gender differences in solving problems in primary school mathematics in the Netherlands. In T. Nunes and P. Bryant (Eds.), *Learning and Teaching Mathematics: An International Perspective*, 223-253. Psychology Press, UK, 1999.

[23] Van Slyke, C. Comunale, C., and Belanger, F. Gender differences in perceptions of web-based shopping. *Communications of the ACM* 45(8), August 2002, 82-86.

[24] von Hellens, L. and Nielsen, S. Australian women in IT. *Communications of the ACM* 44(7), July 2001, 46-52.

[25] Werner, L., Hanks, B., and Mcdowell, C. Pair-programming helps female computer science students, *ACM Journal of Educational Resources in Computing* 4(1), March 2004, 1-8.