Technical Report # 04-60-06 Oregon State University April 2, 2004

Gender: An Important Factor in End-User Programming Environments?

Laura Beckwith and Margaret Burnett
Oregon State University
Corvallis, Oregon 97331
{beckwith, burnett}@cs.orst.edu

Abstract

A human-centric issue that has not been considered in the design of end-user programming environments is whether gender differences exist that are important to the design of these environments. Ignoring this issue would miss the opportunity of enhancing the effectiveness of end-user programmers by incorporating appropriate mechanisms to support gender-associated differences in decision making, learning, and problem solving. This paper takes a first step toward building a foundation for investigating this issue by surveying gender difference literature from five domains from the perspective of possible implications for end-user programming. We present a taxonomy of this literature, and derive a number of specific issues for each element of the taxonomy (stated as hypotheses). This foundation provides a starting point for organized investigations into issues that may be important for making breakthroughs in the effectiveness of end-user programmers.

1. Introduction

Although gender differences in a technological world are receiving significant research attention, much of the research and practice has aimed at how society and education can impact the successes and retention of female computer science professionals. The possibility of gender issues *within software*, however, has received almost no attention, nor has the population of end-user programmers. We hypothesize that gender-related factors within a software environment that supports end-user programming may have a strong impact on how effective end-user programmers can be in that environment. Evidence from other fields and preliminary investigations of our own have already begun to reveal evidence supporting this hypothesis.

Despite substantial human-centric research relevant to end-user programming (e.g. [4, 18, 31, 34]), few researchers have considered potential *gender HCI issues*: gender differences that may need to be accounted for in designing end-user programming environments. The most notable exception is Czerwinski's pioneering research on the support of both genders in navigating through 3-D environments [12, 40]. Evidence from Czerwinski's work as well as work in other domains, such as psychology and marketing, suggests that women process information and problem solve in different ways than men do. In fact, some research has shown that some software is (unintentionally) designed for men [21].

Note that individual differences, such as in learning styles or spatial abilities, are known to have greater effects on an individual's performance than any group-based influences, such as gender or ethnicity. Still, important subsets of these differences tend to cluster by gender, and thus investigating gender-related issues can serve large numbers of people whose differences follow common "clustering" patterns

One reason it is important to consider gender HCI issues in end-user programming is simply that ignorance of these issues is risky. Ignorance of gender issues has already proven to be dangerous: today's low percentage of computer science females [9] has been directly attributed to the past unawareness of gender issues in computer science education and in the workforce. There is a risk that if gender HCI issues of end-user programming environments are ignored, a similar phenomenon could occur with end-user programmers.

This paper surveys literature across disciplinary boundaries to provide an initial foundation for the following open human-centric question: *Is gender an important factor in end-user programming environments?*

This paper's contributions are fourfold: (1) the open question itself, (2) a survey of the relevant literature, (3) a taxonomy of gender issues that satisfy two constraints: they potentially impact end-user programmers' success while at the same time being addressable *within* end-user programming software systems, and (4) specific hypotheses relating to each element of the taxonomy.

2. What Could Go Wrong?

What gender differences might matter in the design of end-user programming environments? Consider the following scenario in one particular end-user programming environment.

Imagine a female teacher engaged in preparing a spreadsheet to track her students' scores and to calculate

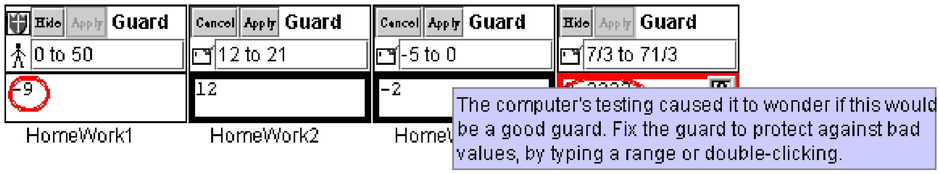

Figure 1: A spreadsheet calculating the average of three homework scores. Assertions about the ranges and values are shown above each cells' value. For example, on HomeWork1 there is a user-entered assertion (noted by the stick figure) of 0 to 50. The other three cells have assertions "guessed" by the Surprise-Explain-Reward strategy. Since the value in HomeWork1 is outside of the range of the assertion, a red circle notifies the user of the violation. A "tool tip" (lower right) shows the explanation for one of the guessed assertions.

ways of providing students with the best grades. Part of her process of preparing her spreadsheet is to test the spreadsheet. While she is engaged in testing, the system surprises her by decorating some of the spreadsheet cells.

The surprises were intentionally placed into the software by the designers relying on a strategy for end-user programming environments called *Surprise-Explain-Reward* [43]. The surprise, which was intended to capture the teacher's attention and arouse her curiosity, reveals the presence of an "information gap" [24]. In this case the system is using the surprise to interest her in assertions [6], which she can use to guard against future errors by specifying, for example, that the value of a cell calculating a grade average should always fall between 0 and 100.

What could go wrong in surprising the user? According to Lowenstein's information gap theory, a user needs to have a certain level of confidence in order to reach a useful level of curiosity [24]. However, given documented gender differences in computer confidence, the teacher's level of computer confidence could interfere with the surprise's ability to capture her interest.

Returning to our scenario, suppose for this particular user, the surprise is effective at arousing her curiosity; she looks to the object that surprised her (the assertion) for an explanation. The explanation, viewed through a tooltip, includes the semantics, possible actions she can take (regarding the assertion), and the future reward(s) of taking the action. (See Figure 1.)

What could go wrong with the explanation? According to one theory, males and females process information differently [28], and thus both the presentation and the content of the explanation may impact its effectiveness for males versus females. If the information needed by the user is not effectively communicated, the user's ability to problem solve is likely to be reduced.

Another role of the explanation is to help users make a reasonably accurate assessment of the risk in taking some action – but since males and females differ in their perceptions of risk, the explanation may need to serve these two populations differently in this respect as well. (An example of risk may be the fear that the user will lose their work if they try a certain feature.) If one gender perceives an explanation of a feature as communicating higher levels of risk than a another, the users with higher risk perceptions may avoid supposedly "risky" features that may be important to overall effectiveness.

Perhaps the most important role of explanations is to make clear the rewards of using particular end-user programming features. Providing information about rewards in the explanation is consistent with the implications of the Model of Attention Investment [4], an analytic model of user problem-solving behavior that models the costs, benefits, and risks users weigh in deciding how to complete a task. An implication of this model is that if the system provides the user an idea of future benefits, users can better assess if the cost of using a feature (here assertions) is worth their time. The reward aspect of the strategy refers to rewards such as the automatic detection of errors, which is depicted by the red circle around HomeWork1's erroneous value in Figure 1.

What could go wrong with rewards? Since males and females are often motivated by different factors, there may be gender differences in what actually is a perceived "reward." If the rewards are only tailored to one gender's perceptions of rewards, the other gender may not be motivated to use the devices that will help them be effective.

In this end-user programming scenario, potential problems arose *that may be addressable within the end-user programming software itself*. Four issues that arose here were (1) software features whose effects depend upon users' computer confidence, (2) the software's ability to communicate effectively with users, (3) the possibility of a user's perception of risk interfering with the user choosing to use appropriate features, and (4) possible differences between a user's actual motivations and the software's attempt to "reward" users for using particular features. These and other issues potentially addressable within end-user software environments will be discussed further in Sections 4-6.

3. A Taxonomy of Gender Differences

Literature on gender differences that may have implications for end-user programming spans a multitude of research areas. The research from within computer science has come primarily from computer science

Issue	Domain contributing evidence	Summary	Potential impacts of these differences on end-user programmers
Confidence Lack of self-confidence Overconfidence Perceived Risk	Computer Science Psychology Education	Females tend to be less confident and to perceive more risk than males do [7, 8, 16, 19, 22, 27].	Users' level of engagement with the software.
Support Learning styles Problem-solving style Information processing	Education Computer Gaming Marketing	Females and males not only have differences in learning styles, they also process information differently [1, 13, 20, 28, 29].	Users' ability to understand communications and problem-solve effectively.
Motivation Ease of use vs. usefulness Technology use	Computer Science Psychology	Males and females are motivated to use technology for different reasons [5, 26, 42].	Users' interest in pertinent end-user programming features that can aid effectiveness in problem solving.

Table 1: Taxonomy of gender differences relevant to end-user programming.

education and the area of computer games. In addition to these areas, we have also drawn research from psychology, marketing, and education.

To organize the relevant literature into a coherent form, we have devised a taxonomy of this work. See Table 1. Each element of the taxonomy is a particular issue that has emerged as a recurring theme from the literature we survey here. Many of the issues in the taxonomy have interdependencies with one another (e.g., confidence can affect motivation and even vice versa), but each issue was selected due to having received attention as a theme in multiple literature works. The next three sections of this paper survey the relevant literature following the sequence of the taxonomy, deriving specific hypotheses along the way.

4. Confidence

This document uses the term "confidence" for the interrelated concepts of self-confidence, self-efficacy, overconfidence, and perceived risk.

4.1 Lack of Self-Confidence

> "I'm actually kind of discouraged now. Like I said before, [there are] so many people who know so much more than me, and they're not even in computer science. Like I was talking to this one kid, and …oh my God! He knew more than I do. It was so… humiliating kind of, you know? So I get discouraged by things like that—I don't know what I think I need to know. And that inhibits my willingness to continue (laughs) … if you can understand that. It shouldn't. It should like make me want to learn even more. But I feel like I'll always be behind, and it's discouraging." [27]

From the field of computer science, there is substantial evidence of low confidence levels as computer science females compare themselves to the males [26, 27]. Of particular pertinence to end-user programming, however, is the evidence showing that low confidence relating to technology is not confined to computer science females [7, 11, 14, 21, 41]. For example, Busch reports on a study with business students who were finishing a year long course covering the use of word processing and spreadsheet software [7]. The study investigated how males and females viewed their ability to complete complex tasks in these programs, and found that females were significantly less confident than the males.

As a measure of confidence, researchers often use *self-efficacy*, as was done in the Busch study. Self-efficacy is belief in one's capabilities to perform a certain task [2, 44]. Pajares argues that self-efficacy can affect task effort, persistence, expressed interest, and the level of difficulty of goals users will strive to attain [33]. Further, there is specific evidence that low self-efficacy impacts attitudes toward a new software package prior to its use [19]. Females' low self-efficacy is further complicated by their tendency to attribute failure at a task to their own lack of capability, whereas the males attribute this to the difficulty of the task [39]. Taken together, this research suggests that a first experience with end-user programming software can impact attitudes which may in turn impact users' future choices to (or not to) use end-user programming features.

This leads to our first hypothesis:

Hypothesis H1: There will be gender differences in users' interest in exploring new features in end-user programming environments.

4.2 Overconfidence

At the opposite end of the confidence spectrum from low self-efficacy is the issue of overconfidence. Overconfidence about one's own performance is a well-known and robust finding in behavioral science research; Panko's survey of a number of findings from multiple domains has helped to document its pervasiveness [35]. In particular, overconfidence in spreadsheet correctness is common [35]. Overconfidence matters because it can prevent people from suspecting errors, leading to misplaced trust in erroneous programs. In our research we have incorporated methods into our end-user programming environment that effectively help to reduce overconfidence in spreadsheet correctness [6], but we do not know how our strategies impact underconfident users.

There is evidence suggesting gender differences in overconfidence just as in underconfidence. For example, in Lunderberg et al.'s work, although both men and women were often overconfident, men were significantly more overconfident in their incorrect answers for math-based computational skills [25]. Pulford and Colman also reported that men were significantly more overconfident than the women [36]. Hence, designing methods to help alleviate overconfidence in end-user programming needs to be carefully targeted specifically toward overconfident users.

Hypothesis H2: In order to overcome the documented tendency of overconfidence by males, it will be especially important to point out potential errors in a manner that is effective for males.

4.3 Perceived Risk

According to the Attention Investment Model (a model of how users allocate their attention in problem solving) [4], a user, in deciding to take any action, first weighs the perceived costs, pay-offs, risks, and benefits of taking that action. If they decide that the costs and/or risks are too high in relation to the benefits they may choose not to follow through with the action. Perception of risk thus plays an important role in a user's decision making about whether to use end-user programming features.

There is evidence that women perceive higher risk in everyday choices and behaviors than men do (especially Caucasian men) [16]. Although Finucane et al.'s research does not consider computer use, it is consistent with findings from other research [3].

Just as the Attention Investment Model predicts, higher perception of risk can lead to differences in actual behavior, and such differences have been tied to gender. For example, researchers have found that women are more risk-averse in their financial decisions [22]. Women are also more risk-averse in "informed guessing" [8]. Informed guessing is the willingness to make an educated guess on questions when the result of an incorrectly answered question is negative (such as losing points versus winning points with correct answers).

These findings may have important HCI implications for end-user programming environments. For example, if women perceive the risk of taking an action regarding some aspect of the software as being higher than men do, and their perceived risk results in avoidance behavior (not using features that might help them in their task), then the result could be a less robust program, or a higher cost (in time) to create the program.

Hypothesis H3: Females' high perceptions of risk will render them less likely to make use of unfamiliar devices in end-user programming environments.

4.4 Preliminary Empirical Work

To obtain evidence about whether pursuing confidence might yield insights into gender HCI issues for end-user programming environments, we conducted a preliminary think-aloud study.

Our preliminary study investigating gender differences duplicated the Surprise-Explain-Reward study reported in [43], except that we switched to a think-aloud protocol [15] and the participants had less spreadsheet experience. To measure pre-task confidence we used a self-efficacy questionnaire that has been used successfully by other researchers before [10]. The task itself was to debug two spreadsheets, during which assertions were introduced to the participants via the Surprise-Explain-Reward strategy described in Section 2. To measure post-task confidence, the post-task questionnaire asked the participants to rate on a scale of 1 to 5 how confident they were that they had found and fixed all the bugs. To our surprise, the females' confidence levels dropped *over the course of the study* much more than did the males' confidence levels. (The difference in the drop between the two genders was nearly significant at the .05 level: $p=0.0513$, despite having only 13 participants in the study.) This result suggests that the end-user programming environment itself—which, like all other end-user programming environments, was designed without knowledge of gender-related HCI principles—is not currently serving the females' needs as well as the males'.

5. Support

We will use the term "support" to mean the built-in aspects of the software that help the users learn or understand the environment better. Two examples of support aspects are on-line help systems (to explain features in a software package), and devices within the environment which provide explanations, such the tooltips of Figure 1. Here we consider gender differences that have possible implications on the types of support males and females need when working in end-user programming environments.

5.1 Learning Style

Even when end users have some formal training in a particular software system, they are unlikely to remember all of the software's devices. Thus, the system must include devices to help the users achieve this mastery. On-line help systems are a prime example.

The system's approach to help users achieve such mastery may depend on a user's learning style. One survey of university students found that students with an "abstract random" learning style were both significantly more likely to be female and had a negative correlation with computer confidence [1]. (In this study users who could mentally manipulate formless concepts were "abstract", while the "random" aspect is characterized as non-linear thinkers.) The author concluded that females using a predominantly abstract random learning style may find computer-based instruction ineffective for learning [1].

Other researchers have also found gender differences in learning styles [13, 20, 37], but many of these studies show only small gender differences, and at times contradict one another. However, one "bottom line" from this work is that the end-user programming field may need to consider several learning styles when designing support devices, especially if some users are easily dissuaded by support that is not sensitive to their learning style.

Hypothesis H4: Gender differences in learning style will cause some software devices aiming to "teach" new features or procedures to be less effective for one gender than another.

5.2 Problem-Solving Style

We have just considered how learning style may impact the software's ability to teach (and therefore support) use of its features. Another aspect of a support system's job is to teach problem solving using the features. However, there are gender differences in the way people solve problems.

Several gender difference pertinent to problem solving have been uncovered by research in computer games. To investigate possible reasons girls play fewer computer games than boys, Kafai examined the differences in the games that girls and boys designed [23]. When asked to create an educational game, girls and boys created games with distinguishing differences. For example, girls are more likely to create games with a "teaching" theme, while the boys designed games built on adventure.

Furthermore, other researchers found that boys and girls prefer to work through games in different ways. Rather than working in a linear fashion through the game girls prefer to explore and move freely about a game. (These findings are summarized in [17].) If these results translate to the ways females problem solve in end-user programming, this could imply the software needs to support both genders through a design which allows for linear and non-linear problem-solving styles.

Hypothesis H5: An end-user programming environment that restricts users to a linear (non-linear) approach will adversely impact females' (males') abilities to problem-solve effectively in that environment.

In another difference in boys' problem-solving style versus girls': each gender prefers different interaction techniques in games. Boys' games typically depend upon competition, whereas girls prefer collaboration and working together. In end-user programming environments these problem-solving strategies could imply differences in the support provided by the system. For example, providing support for collaboration may be important to females. (In fact, there is evidence from within the end-user programming domain that collaboration is a common technique used by end-user programmers [31].)

Hypothesis H6: An end-user programming environment that explicitly supports collaboration will enable females to complete their tasks more effectively than environments without this support.

5.3 Information Processing

In addition to gender differences in learning style and problem-solving style, there is another gender-related factor that affects whether a user can benefit from an environment's support devices: how the user processes the information the support devices provide.

Research in the area of marketing has turned up a variety of gender differences relevant to information processing [28, 29, 38]. One of these works, the theory of selectivity [28, 29], is particularly suggestive. The theory states that males and females differ in their information processing strategies. For advertisers, these differences are important in designing ads to reach specific populations. For end-user programming environments, this research may have implications for informing end users of important information via the software's support devices.

According to the theory of selectivity, females are more likely to employ elaborative information processing strategies, regardless of whether the task is simple or complex in nature [28, 32]. Males, however, are more likely to select heuristic processing strategies that minimize cognitive effort and reduce information load for simple tasks, switching to an elaborative strategy only on more complex tasks [32].

These gender differences have been shown to impact diverse software-related activities, ranging from users' perceptions of web sites used for e-commerce [38] to users' performance on auditing tasks [32].

Hypothesis H7: Males will be less likely than females to thoroughly read complicated or lengthy "help" explanations.

6. Motivation

6.1 Views of Technology

Computer science research has shown that computer science females are motivated by how technology can help other people, whereas males tend to enjoy technology for its own sake [26]. The following quote is from a female at Carnegie Mellon, describing why she chose to major in computer science:

> "I think with all this newest technology there is so much we can do with it to connect it with the science field, and that's kind of what I want to do (study diseases) … Like use all this technology and use it to solve the problems of science, the mysteries" [26].

These differences are also are representative of other women who use technology, such as architects, NASA scientists, and filmmakers. In one study [5], women and men were asked to envision writing a scientific story in which they used a technological object, and were then asked to describe this object. The women described objects as tools to help integrate personal and professional lives and to facilitate creativity and communication. The men's descriptions, however, used the technological device to increase command and control over nature and one another. The distinctions between how men and women viewed technology are summarized in Table 2.

Table 2 directly suggests several hypotheses:

Hypothesis H8 (suggested by row 1): End-user programming environments that support "productizing" a user's program (for example by allowing a user to keep his "source code" private) will be perceived to be more valuable by male end-user programmers than environments that do not have these features.

Hypothesis H9 (suggested by row 3): End-user programming environments that support communication (for example by connecting users working on similar environments though the network) will be perceived to be more valuable by female end-user programmers than environments that do not have these features.

Hypothesis H10 (suggested by row 8): End-user programming environments that support sharing will be perceived to be more valuable by female end-user programmers than environments that do not have these features.

	Women …	**Men …**
1	Fantasize about it as a **medium**	Fantasize about it as a **product**
2	See it as a **tool**	See it as a **weapon**
3	Want to use it for **communication**	Want to use it for **control**
4	Are impressed with its potential for **creation**	Are impressed with its potential for **power**
5	See it as **expressive**	See it as **instrumental**
6	Ask it for **flexibility**	Ask it for **speed**
7	Are concerned with its **effectiveness**	Are concerned with its **efficiency**
8	Like its ability to facilitate **sharing**	Like its ability to facilitate **autonomy**
9	Are concerned with **integrating** into their personal lives	Are intent on **consuming** it
10	Talk about wanting to **explore** worlds	Talk about using it to **exploit** resources and potentialities
11	Are **empowered** by it	Want **transcendence**

Table 2: Summarization of gender differences in ways men and women fantasized about technology [5].

Hypothesis H11: An end-user programming environment that emphasizes the potential usefulness of its features (such as in its on-line help content) will be perceived as being more valuable by male end-user programmers than an environment that does not emphasize usefulness of its features.

6.2 Ease of Use vs. Perceived Usefulness

The Technology Acceptance Model (TAM) [30, 42] provides researchers with a model to examine users' acceptance and usage behavior of technology. According to TAM, user acceptance, and ultimately technology use, is determined by two key beliefs: perceived usefulness and perceived ease of use [42]. "Perceived usefulness" is defined as the degree to which a user believes that using the system will enhance their performance, and "perceived ease of use" is defined as the degree to which the user believes that using the system will be free of effort. According to one study, the relative importance of each differs by gender [42]; they found that women were more influenced by perceived ease of use in adapting new technology whereas men were more strongly influenced by perceived usefulness.

7. Discussion

In Sections 4-6, we presented the literature basis of each element of the taxonomy, including a specific set of hypotheses suggested by that literature. A complete list of these hypotheses is given in Table 3.

Although the hypotheses are presented in "encapsulated" ways, taking only one issue into account each, some interactions will obviously arise from any decisions that might be made based upon the outcomes of individual investigations into these hypotheses. For example, regarding confidence, the contrasting overconfidence trend for male users, and trend for high risk perceptions by female users make clear that a design decision made to help address confidence-related issues for one gender might well have exactly the wrong effect on the opposite gender.

In some cases, the hypotheses that we derived are also able to directly suggest design approaches. For example, in the support category, consider hypothesis H5. If formal investigation into this hypothesis verified that it was indeed well supported, the obvious solution would be for an end-user programming environment to avoid *restricting* users to a single problem-solving style, instead giving them the flexibility to choose either (or both) as needed.

Confidence
H1: There will be gender differences in users' interest in exploring new features in end-user programming environments.
H2: In order to overcome the documented tendency of overconfidence by males, it will be especially important to point out potential errors in a manner effective for males.
H3: Females' high perceptions of risk will render them less likely to make use of unfamiliar devices in end-user programming environments.
Support
H4: Gender differences in learning style will cause some software devices aiming to "teach" new features or procedures to be less effective for one gender than another.
H5: An end-user programming environment that restricts users to a linear (non-linear) approach will adversely impact females' (males') abilities to problem-solve effectively in that environment.
H6: An end-user programming environment which explicitly supports collaboration will enable females to complete their tasks more effectively than environments without this support.
H7: Males will be less likely than females to thoroughly read complicated or lengthy "help" explanations.
Motivation
H8: End-user programming environments that support "productizing" a user's program (for example by allowing a user to keep his "source code" private) will be perceived to be more valuable by male end-user programmers than environments that do not have these features.
H9: End-user programming environments that support communication (for example by connecting users working on similar environments though the network) will be perceived to be more valuable by female end-user programmers than environments that do not have these features.
H10: End-user programming environments that support sharing will be perceived to be more valuable by female end-user programmers than environments that do not have these features.
H11: An end-user programming environment that emphasizes the potential usefulness of its features (such as in its on-line help content) will be perceived as being more valuable by male end-user programmers than an environment that does not emphasize usefulness of its features.

Table 3: Summary of the hypotheses derived for each element of the taxonomy.

8. Conclusion

Drawing from five domains, this paper has surveyed literature relevant to the following open question:

Is gender an important factor in end-user programming environments?

All of the literature included in this survey meets the following constraints: it identifies one or more issues that (1) potentially impact end-user programmers' success and (2) are potentially addressable *within* end-user programming software systems.

From the surveyed literature, we proposed a taxonomy, each of whose elements is a recurring theme in the literature. Using the taxonomy as an organizing device, we then derived a number of subquestions, stated as hypotheses about end-user programming environment characteristics, that are strongly suggested by the literature. Together, the open question, survey, taxonomy, and hypotheses are intended to provide a foundation for future investigation into this human-centric programming issue.

Acknowledgments

We thank the members of the Visual Programming Research Group at Oregon State University for their feedback and help. This work was supported in part by Microsoft Research and by the EUSES Consortium via NSF grant ITR-0325273.

References

[1] Ames, P., Gender and learning styles interactions in student's computer attitudes, *J. Educational Computing Research*, 28(3), 2003, 231-244.

[2] Bandura, A., Self-efficacy. In V. S. Ramachaudran (Ed.), *Encyclopedia of Human Behavior* (Vol. 4). New York: Academic Press, 1994, 71-81.

[3] Barke, R., Jenkins-Smith, H., and Slovic, P., Risk perceptions of men and women scientists, *Social Science Quarterly,* 78(1), 1997, 167-176.

[4] Blackwell, A., First steps in programming: a rationale for Attention Investment models. *IEEE Human-Centric Computing Languages and Environments,* 2002, 2-10.

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

[6] Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J., and Wallace, C., End-user software engineering with assertions in the spreadsheet paradigm, *Proc. Int. Conf. Software Engineering,* 2003, 93-103.

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

[8] Byrnes, J., Miller, D., and Schafer, W., Gender differences in risk taking: a meta-analysis, *Psychological Bulletin,* 125(3), 1999, 367-383.

[9] Camp, T., The incredible shrinking pipeline, *Comm. ACM,* 40(10), 1997, 103-110.

[10] Compeau, D., and Higgins, C., Computer self-efficacy: development of a measure and initial test, *MIS Quarterly,* 19(2), June 1995, 189-211.

[11] Cottrell, J., I'm a stranger here myself: a consideration of women in computing, Spec. Interest Grp. User Services Conf. XX, ACM Press, 1992, 195-202.

[12] Czerwinski, M., Tan, D., and Robertson, G. Women take a wider view, *ACM Conf. Human Factors in Computing Systems,* ACM Press, 2002, 195-202.

[13] de Lange, P., and Mavondo, F., Gender differences in learning styles and quality of learning among accounting students, *Accounting Association of Australia and New Zealand Conference*, 2002. http://www.aaanz.org/web2002/accepted%20_papers/delangep.pdf

[14] Durndell, A., and Haag, Z., Computer self efficacy, computer anxiety, attitudes towards the internet and reported experience with the internet, by gender, in an Eastern European sample, *Computers in Human Behavior,* 18, 2002, 521-535.

[15] Ericsson, K. and Simon, H., *Protocol Analysis*, Cambridge, MA: MIT Press, 1984.

[16] Finucane, M., Slovic, P., Mertz, C., Flynn, J., and Satterfield, T., Gender, race, and perceived risk: the 'white male' effect, *Health, Risk & Society,* 2(2), 2000, 159-172.

[17] Gorriz, C., and Medina, C., Engaging girls with computers through software games, *Comm. ACM,* 43(1), 2000, 42-49.

[18] Green, T. R. G. and Petre, M. Usability analysis of visual programming environments: a 'cognitive dimensions' framework, *J. Visual Languages and Computing,* 7(2), June 1996, 131-174.

[19] Hartzel, K., How self-efficacy and gender issues affect software adoption and use, *Comm. ACM,* 43(9), 2003, 167-171.

[20] Heffler, B., Individual learning style and the learning style inventory, *Educational Studies*, 27(3), 2001, 307-316.

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

[22] Jianakoplos, N. and Bernasek, A., Are women more risk averse? *Economic Inquiry,* Oct. 1998, 620-630.

[23] Kafai, Y., Video game design by girls and boys: variability and consistency of gender differences, In J. Cassell and H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and Computer Games*, Cambridge, MA: MIT Press, 1998, 90-114.

[24] Lowenstein, G., The psychology of curiosity, *Psychological Bulletin* 116(1), 1994, 75-98.

[25] Lunderberg, M., Fox, P., and Punchochar, J., Highly confidence but wrong: gender differences and similarities in confidence judgments, *J. Educational Psychology,* 86(1), 1994, 114-121.

[26] Margolis, J., Fisher, A., and Miller F., Caring about connections: gender and computing, *IEEE Technology and Society,* 18(4), December 1999.

[27] Margolis, J., Fisher, A., and Miller F., Living among the "programming gods": the nexus of confidence and interest for undergraduate women in computer science (work in progress – CMU). http://www-2.cs.cmu.edu/~gendergap/confidence.html

[28] Meyers-Levy, J., Gender differences in information processing: a selectivity interpretation, in *Cognitive and Affective Responses to Advertising,* Lexington Books, 1989, 219-260.

[29] Meyers-Levy, J. and Sternthal, B., Gender differences in the use of message cues and judgments, *J. Marketing Research,* Feb. 1991, 84-96.

[30] Morris, M. and Dillon, A., The influence of user perceptions on software utilization: application and evaluation of a theoretical model of technology acceptance, *IEEE Software,* 14(4), 1997, 58-76.

[31] Nardi, B. *A Small Matter of Programming: Perspectives on End-User Computing,* Cambridge, MA: MIT Press, 1993.

[32] O'Donnell, E. and Johnson, E., Gender effects on processing effort during analytical procedures, *Int. J. Auditing,* 5, 2001, 91-105.

[33] Pajares, F., Gender and perceived self-efficacy in self-regulated learning, *Theory Into Practice,* 41(2), 2002, 116-125.

[34] Pane, J., Ratanamahatan, C., and Myers, B., Studying the language and structure in non-programmers' solutions to programming problems, *Int. J. Human-Computer Studies,* 54(2), 2001, 237-264.

[35] Panko, R., What we know about spreadsheet errors, *J. End User Computing,* Spring 1998.

[36] Pulford, B., and Colman, A., Overconfidence: feedback and item difficulty effects, *J. Personality and Individual Differences,* 23(1), 1997, 125-133.

[37] Severiens, S., and ten Dam, G., Gender and gender identity difference in learning styles, *Educational Psychology*, 17(1/2), 1997, 79-93.

[38] Simon, S., The impact of culture and gender on web sites: an empirical study, *The DATA BASE for Advances in Information Systems,* 32(1), 2001, 18-37.

[39] Stipek, D., and Gralinski, J.H., Gender differences in children's achievement-related beliefs and emotional responses to success and failure in mathematics, *J. of Educational Psychology*, 83(3), 1991, 361-371.

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

[41] Torkzadeh, G., and Van Dyke, T., Effects of training on Internet self-efficacy and computer user attitudes, *Computers in Human Behavior,* 18, 2002, 479-494.

[42] Venkatesh, V., and Morris, M., Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior, *MIS Quarterly,* 24(1), 2000, 115-139.

[43] Wilson, A., Burnett, M., Beckwith, L., Granatir, O., Caasburn, L., Cook, C., Durham, M., Rothermel, G., Harnessing curiosity to increase correctness in end-user programming, *ACM Conf. Human Factors in Computing System,* ACM Press, 2003, 305-312.

[44] Zeldin, A. and Pajares, F., Against the odds: self-efficacy beliefs of women in mathematical, scientific, and technological careers, *American Educational Research Journal*, 37, 2000, 215-246.