D. (METHODOLOGY TRACK): QUANTITATIVE ANALYSIS OF SOCIAL STRUCTURES

Gender and Programming Achievement in a CSCL Environment

Amy Bruckman, Carlos Jensen
Georgia Institute of Technology
{asb,carlosj}@cc.gatech.edu

Austina DeBonte
Microsoft
austinad@microsoft.com

ABSTRACT
In this study, we analyzed 3.4 GB of log file data from the participation of 475 children in a CSCL environment over a period of five years. Using scripts to divide the children's commands typed into categories, we found that girls spend significantly more time than boys communicating with others in the CSCL environment. Analyzing the children's level of programming achievement, we found that gender does not affect programming performance. Regression analysis shows that performance is correlated with prior programming experience and time on task. Boys are more likely than girls to have prior programming experience, and spend more time programming on average. We contrast these quantitative findings with our qualitative observations, and conclude that quantitative analysis has an important role to play in CSCL research. These results suggest that educators wishing to increase gender equity in technical skill should focus on strategies for fostering interest among girls.

Keywords
Gender, programming, learning, CSCL

A GENDER GAP?
Since the earliest days of the personal computer (and perhaps earlier), researchers have been asking questions about gender equity in computer use. "If males and females participate differentially in computer learning environments, this could lead to differences in cognitive attainments and career access," wrote Marcia Linn in 1985 (Linn, 1985). Linn studied organized middle-school programming classes, and found that girls and boys have similar levels of programming achievement once they enroll in classes, but that girls are less likely to enroll. In the following decade a number of researchers have studied gender differences in how children learn and use computers both in schools and at home (for overviews see Bannert 96 and Giacquinta 93) and the reasons behind these differences (Turkle 86). A number of studies have shown that girls tend to have less exposure to computers. Studies have shown a strong positive relationship between prior experience and both attitudes towards computers and achievement (Kersteen 88, Shashaani 94). When gender role identity and previous experience are accounted for, boys and girls perform equally well (Colley 94).

Since Marcia Linn did her first study, much has changed about computers, access to technology, and how computers are used at home as well as schools. However, the basic facts of gender and computing for kids have not changed: our results replicate Linn's early findings. Most of the work to date has been done using surveys and attitudinal inventories. In this study, we use quantitative log file analysis as well as qualitative observations and interviewing to study student programmers in a mixed school and free-time use environment. While boys develop significantly more programming expertise than girls (25% difference, p=.004), regression analysis shows that in fact this difference is attributable to the fact that boys chose to spend more time programming in the environment, and are more likely to have prior programming experience. Boys are more likely to chose to program both before and during their exposure to our programming environment, and time on task predicts level of achievement.

BACKGROUND: MOOSE CROSSING
In the spring of 1992, Amy Bruckman presented a paper on the fluidity of identity in text-based virtual worlds (or "MUDs") to a student reading group at the MIT Media Lab. A few days later, Mitchel Resnick, then a graduate student but about to join the faculty, posed a question to Bruckman: would it make sense to create a MUD based on the *Babysitter's Club* series of books to encourage elementary and middle-school girls to be interested in computers? This was the beginning of the MOOSE Crossing project.

A new programming language ("MOOSE") and programming environment ("MacMOOSE") were developed to make it easier for children to learn to program (Bruckman, 1997; Bruckman & Edwards, 1999). (A Windows version of the programming environment, “WinMOOSE,” was developed a few years later.) The *Babysitter* series theme was abandoned in favor of a more open-ended, gender-neutral theme. Rather than create an environment for girls, we decided to create an environment we hoped would appeal to both genders, so that we could compare girls’ and boys’ activities there. However, gender was soon relegated to a lower research priority, because there was simply so much fundamental work to do on the basic nature of learning in this new kind of CSCL environment (Bruckman, 2000; Bruckman, Edwards, Elliott, & Jensen, 2000).

Children began to use MOOSE Crossing in the fall of 1995. Everything typed on MOOSE Crossing is recorded, with written informed consent from parents and assent from children. In January 1996, then undergraduate Austina De Bonte joined the MOOSE Crossing development team as part of MIT’s Undergraduate Research Opportunities (UROP) program. DeBonte (then Austina Vainius) wrote a series of Perl scripts to break down children’s commands typed on MOOSE Crossing into categories (see Table 1).

- Movement in the virtual world
- Communication with others
- Consulting the help system
- Looking at people and objects
- Creation objects
- Seeking information about others
- Scripting
- Manipulating object properties
- Looking at object properties
- Using the in-world mail system

DeBonte analyzed 700MB of MOOSE log file data from the first use by kids in September 1995 until April 1997. A total of 160 children participated during this time. Comparing girls and boys use of each of these categories of commands, she found no significant differences. Girls spent more time communicating with others online, and boys had a slightly higher percentage of their commands typed in other categories; however, none of these differences were significant. At the time, we were disappointed in this result—it seemed uninteresting. It is discussed in a few pages of Bruckman’s PhD thesis (Bruckman, 1997), but was not published elsewhere.

Five years later, we looked back at this result, and saw it in a new light. No significant differences is not a lack of results—it is in fact an interesting finding. Consequently, Carlos Jensen dusted off DeBonte’s Perl scripts, and repeated the analysis on our greatly enlarged data set. Additionally, new scripts have been written to analyze children’s level of programming achievement.

DATA ANALYSIS

Categorizing Activity

Through November 2000, 457 children have now participated in MOOSE Crossing. Of those, some children continue to participate for many years, while others try the environment once and never return (see Table 2). This nearly 3-fold increase in experimental subjects has led to a nearly 5-fold increase in the amount of log file data. As of November 2000, we have 3.4 GB of data (compared to 0.7 GB in 1997).

	BOYS	GIRLS	ALL
Mean	4036.8	4914.9	4444.2
Standard Deviation	8906.2	17642.7	13662.5
Median	580	459	516
Minimum	2	5	2
Maximum	55709	177680	177680

Table 2: Time on Task (Commands Typed) By Gender

In total, 46% of MOOSE users are girls, and 54% are boys. This is little changed from 1997, when 43% were girls. It is also similar to the gender distribution on the Internet as a whole. While men dominated the Internet in the mid-90s, men and women were roughly equally represented online by the turn of the century (Abernathy, 2000).

Participation on MOOSE Crossing is measured by counting the total number of commands typed by a member (see Table 2). Since a user might leave a connection window open without actually being present at the computer, connect time is not a useful metric. Total commands typed is a better measure of degree of participation. Differences in time on task by gender are not statistically significant. A few girls have extremely high participation rates (see Figure 4), leading to the mean commands typed being higher for girls while the median is higher for boys. Given the highly variable nature of participation rates, median values are more indicative than means.

A typical entry in our log files looks like this:

```
16:05:38 #218 #78 >>>> say hi
16:05:40 #78 << You say 'hi'
16:05:40 #99 << Amy says, 'hi'
```

Data is stored in files for each day. Each line of input from the user consists of a timestamp, the unique identifier of the room in which the user is located, the user's object number, and ">>>>", followed by what the user typed. Each line of output presented to a user is proceeded by a timestamp, the user's object number, and then "<<", followed by what the user saw. In the above log, Amy (player #99) is in Ginny's Little Cottage (room #218), and says hi. Ginny (#99) hears her.

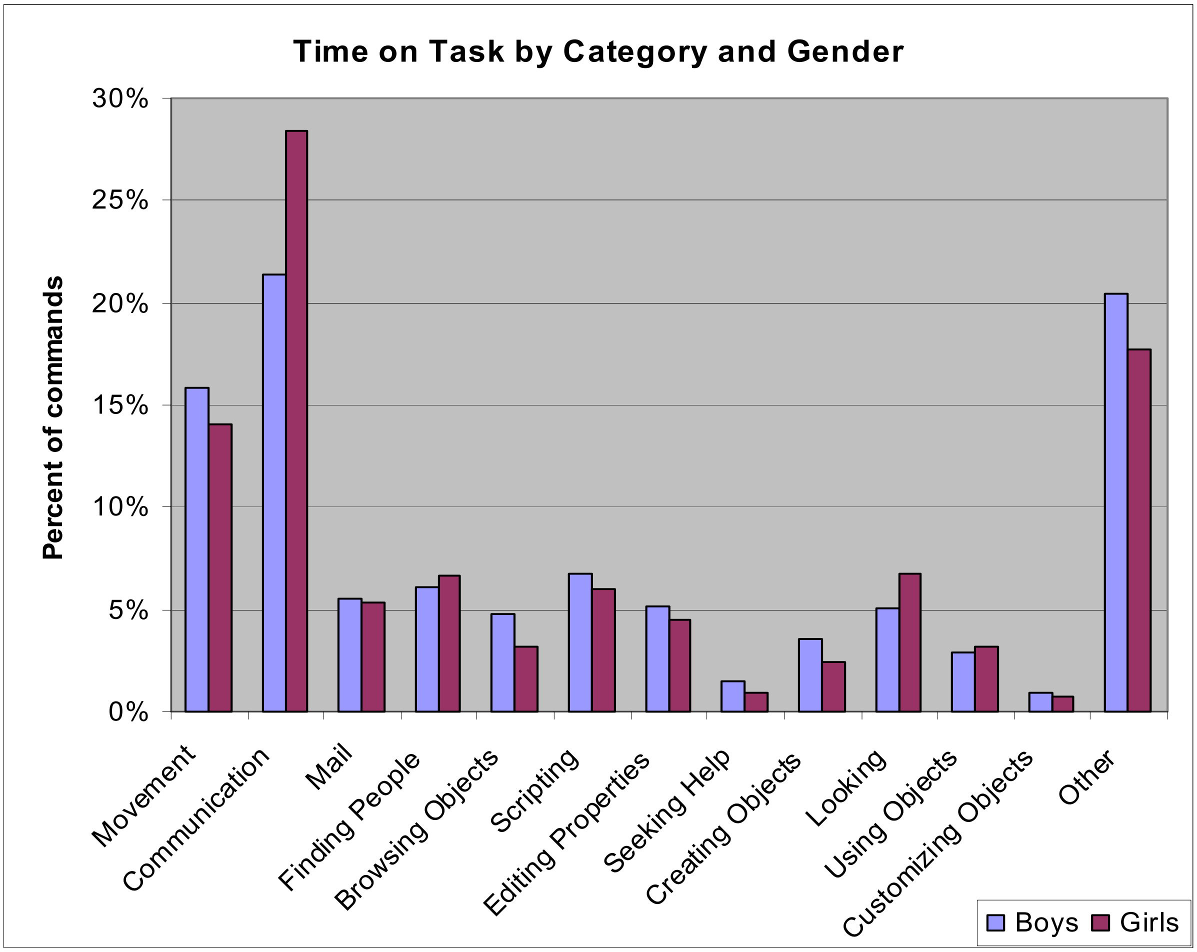

Figure 1

All transactions between client and server are also recorded, allowing us to see when the user looked at a particular object, script, help message, etc. Simple regular expression matching enables us to sort more than 80% of commands typed into categories (see Figure 1).

Only one gender difference in this chart approaches statistical significance: girls spend more time (as a percentage of total commands typed) communicating with others compared to what boys do (marginal significance: p=.082). This trend was also observed in the 1997 data analysis, but at the time was less significant. Time on task as measured by proportion of total commands typed is a zero-sum game. While none of the other differences are significant, the fact that girls are spending more time communicating means they are spending slightly less time than boys in almost all other categories.

We might infer from this that girls appreciate the social nature of the CSCL environment. It is unclear, however, what impact if any this has on girls' learning. Conversation might or might not be contributing to their intellectual growth. We have not analyzed what percent of the communication is "on task" (about programming or writing), versus being purely social. Furthermore, even when communication is purely social, it is not clear to what extent this contributes to the development of writing skills. Thus, this finding is intriguing but difficult to interpret.

Scoring of Student Achievement

In 2000, we used portfolio scoring techniques to analyze students' programming achievement on MOOSE Crossing according to the following scale:

0: Wrote no scripts
1: Demonstrated understanding of basic input/output
2: Used variables and properties
3: Performed list manipulation and flow control
4: Demonstrated mastery of all aspects of the system

These ratings were produced by two human raters. In cases where the raters disagreed, a third person rated the student's level of accomplishment. This technique was applied to a random sample of 50 participants (Bruckman et al., 2000). Subsequently, we became concerned that perhaps our categories were poorly designed. What if kids are learning commands in an unusual order, learning some commands typically classified as "advanced" before others we think of as elementary? Does this set of categories represent student achievement well?

Consequently, we developed a new 100-point achievement scale. Programming commands were divided up into categories: input & output, string manipulation, logic, list manipulation, flow control, and documentation. Most of these categories have sub-categories corresponding to specific commands or concepts. In I/O there are 8 sub-categories, 3 in string manipulation, 4 in logic, 8 in list manipulation, and 8 in flow control. Documentation is the only category that had no sub-elements.

Each kid's scripts were examined for the use of all these elements, and an overall composite score was generated by weighing the different elements according to the importance we assigned to them. Each element in the I/O category was weighted by a factor of 3.5 (28%), strings by 2 (6%), logic by 3 (12%), list manipulation by 2 (16%), flow control by 4 (32%), and documentation by 6 (6%). This gives us an overall score on a scale of 1 to 100.

In fact, our concerns were unfounded: the old and new scores correlate well (see Figure 2). (The comparison is somewhat strained by the fact that the new metric was taken a year later, and some of the students have continued to participate and learn during that year but the others haven't.) Both new and old scales have the limitation that there are relatively subtle differences between categories two and three. One advantage of the new automated technique is that we can analyze all study participants instead of a sub-sample.

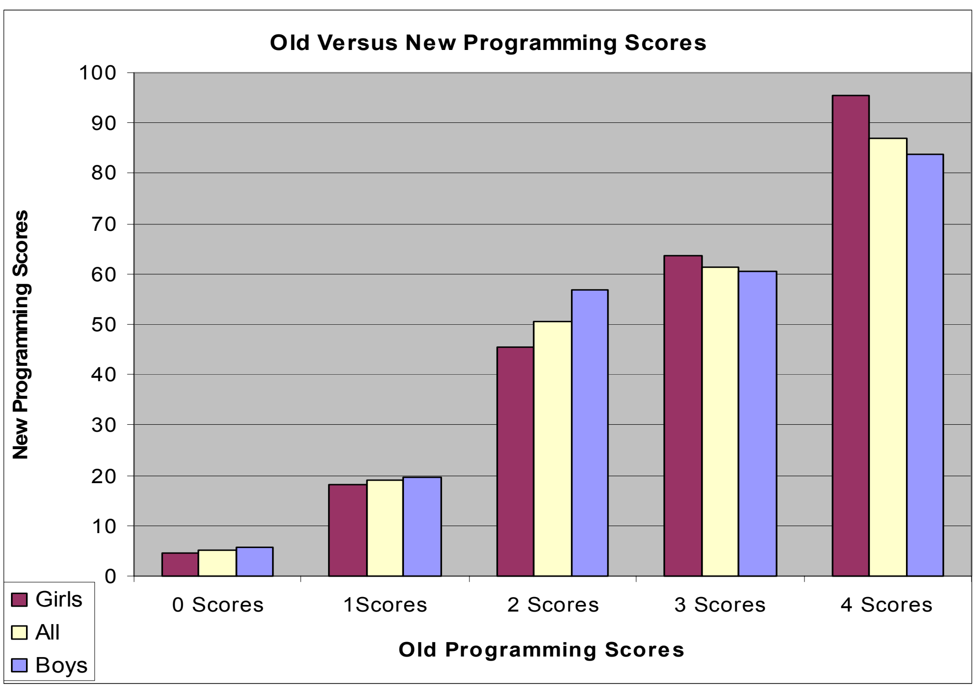

Figure 2

On registering for MOOSE Crossing, all participants are asked if they have previous programming experience. Prior experience is self reported, and generously interpreted. If a student reported any kind of programming experience (for example, having authored HTML), this was counted as an affirmative response. Students with previous experience have significantly higher levels of programming achievement (p=.001) (see Figure 3). The error bars show the extremely large degree of variability in students' achievement. However, despite this variability, the large size of the data set means that the effect is highly significant.

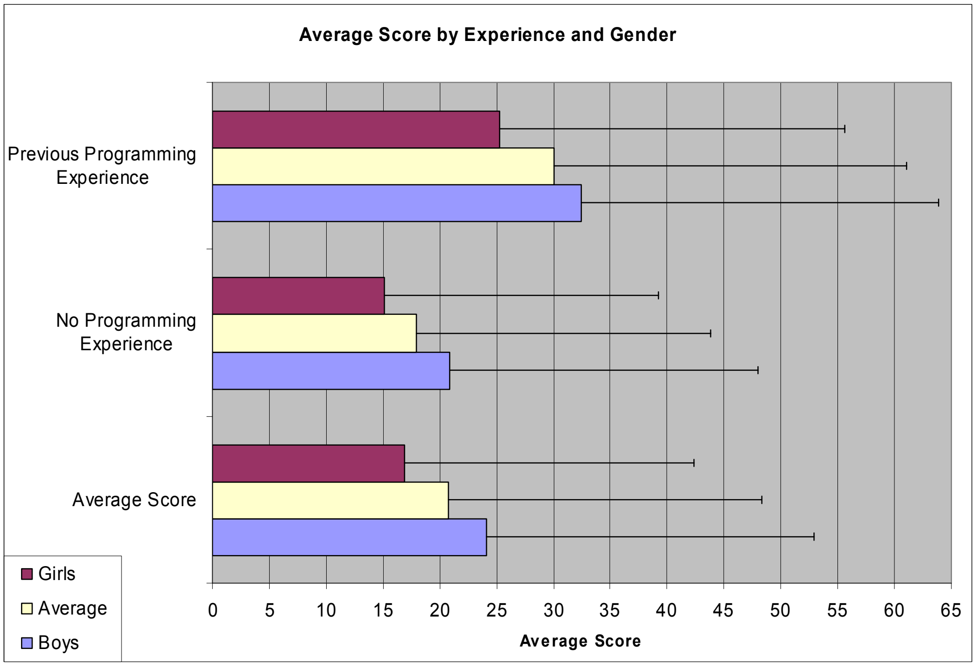

Figure 3

At first glance, boys have a higher level of programming achievement than girls (p=.004). However, regression analysis shows that the difference is explained by prior programming experience (see Figure 3).

Regression analysis is a statistical tool for evaluating the relationship of a set of independent variables to a single dependent variable. This method is particularly useful in situations such as this, where we cannot control the independent variables, yet need to determine their individual effects. Regression analysis seeks to take a set of data-points and find the mathematical equation which best and most reliably describes the given data set. The resulting equation serves as a predictor for the "weight," or "importance" of the different independent variables in relation to the dependent variable, and to each other. In this case, we used a Least Square estimation method, excluding outliers (kids with more than 60,000 commands typed, kids who first used computers after the age of 13, kids who first started using MOOSE after the age of 16, or who spent more than 20% of their total time-on-task programming). Looking at Programming Score, the resulting equation was:

Programming Score = 5.595
+ 3.589 * (if the subject has previous programming experience)
- 0.007 * (time talking to others)
+ 2.93×10^{-7} * (time talking to others)^2
+ 0.006 * (time on task)
- 1.03×10^{-7} * (time on task)^2
+ 0.009 * (time spent scripting)

(Adjusted $R^2 = 0.76$, S.E.=12.8, $p<0.01$ for all except Previous Programming Experience, $p<0.05$).

In other words, programming scores were positively related to time on task, time spent programming (effort), and previous programming experience. Time on Task has a decreasing marginal return (its positive effect plateaus). Programming scores were negatively related to the time they spent talking to others, or engaging in any activity other than programming. (time on task is a zero-sum game). The negative marginal effect of time talking to others is lower the more they talk (it plateaus as it becomes a predictor for continued membership). Gender, environment of use (home, school or other) proved not to be statistically significant.

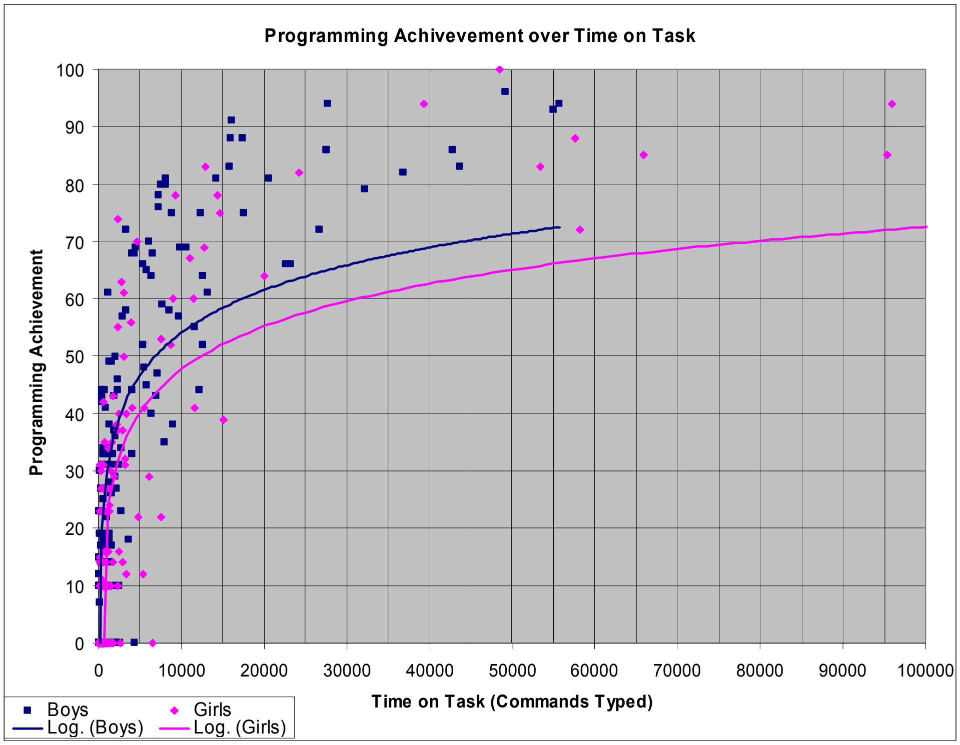

Figure 4

Programming achievement seems most directly related to the time spent in MOOSE as a whole, and more specifically on programming within MOOSE. We therefore chose to look at the factors that determine how much effort the kids put into programming on MOOSE. This resulted in the following equation:

Time spent scripting = 66.42
+ 48.19 (if the subject is a boy)
+ 0.141 * (time talking to others)
- 5.153 $*10^{-6}$ * (time talking to others)^2
+ 3.33 * (help commands)
- 0.003 * (help commands)^2

(Adjusted R^2=0.61, S.E.=233.57, $p<0.05$ for all)

In other words, gender has an effect on the amount of time spent scripting. Interestingly, we find strong evidence for the social nature of MOOSE Crossing and the community support for learning. This can be seen by the fact that communicating with others is a strong indicator for time spent on scripting. The positive marginal effect of time talking to others is lower the higher the level (i.e. at some point communication is not supporting learning, but rather purely social). We also see that consulting the built-in help system has a positive, but marginally decreasing effect (again, the difference between looking up something and randomly accessing help functions). All other factors proved to be statistically insignificant.

Self-Selected Versus Mandatory Use

Home users of MOOSE Crossing are self-selected. On the other hand, school users generally have no choice: they are assigned to participate. It's not surprising, then, that home users have a higher average level of achievement — they have chosen to participate of their own free will, and hence have higher motivation. (This difference is not statistically significant, but the apparent trend is suggestive.) Interestingly, this difference is greater in girls than boys. Boys working from home score on average 1.54 points higher (6.5% higher) than those working from school; girls working from home score on average 4.98 points higher (35% higher) than girls working from school. (These figures are suggestive but not significant.)

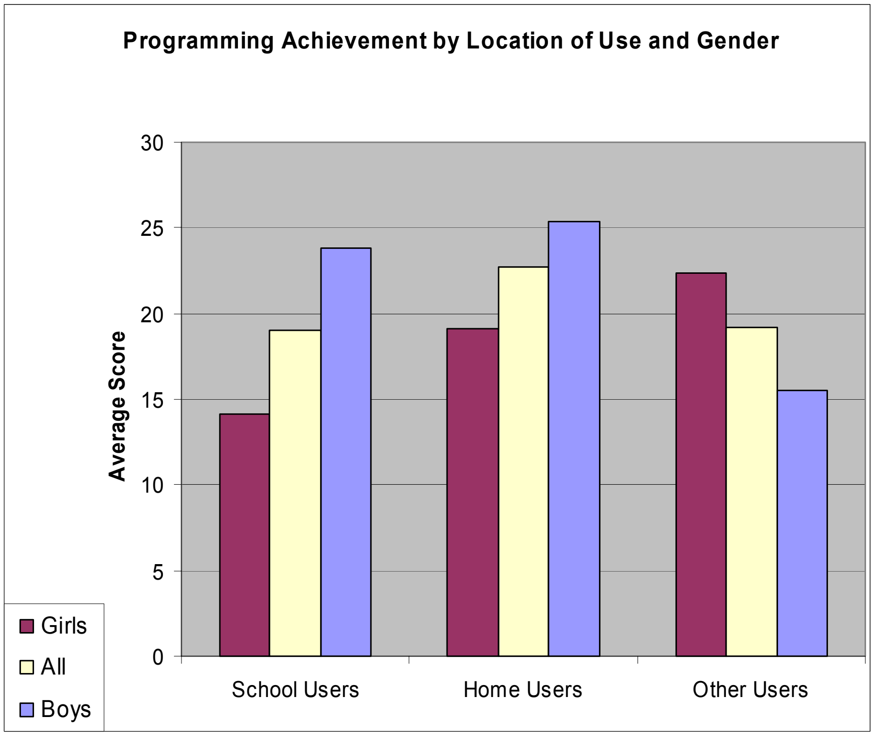

Figure 5

This apparent trend is consistent with our other findings. Overall, girls tend to be less interested in programming than boys. Those girls who self-select to participate are those who happen to be interested. Among the school-use population, girls are less likely to be sincerely interested in the activity. Performance correlates with interest.

Comparison of Quantitative Findings and Informal Observations

Much research in CSCL relies on qualitative data. Sometimes this data is detailed and systematic in its collection; sometimes it is anecdotal. Log file data is easy to collect but difficult to interpret. For example, versions of the Logo programming language generate "dribble files," but few studies have ever made sense of the data in them.

Since 1995, roughly a dozen administrators have spent hundreds of hours each, working with children on MOOSE Crossing. Through those interactions, we necessarily develop informal impressions of the comparative achievement of girls and boys in the environment. The consensus of these impressions is that the achievement of the girls is particularly remarkable, and exceeds that of boys. These impressions turn out to be incorrect. As Figure 4 indicates, the top five participants in terms of total commands typed are girls. A disproportionate amount of our interactions with users are with these dedicated regulars, and this skews our impressions. Quantitative analysis forms a clearer picture. Quantitative analysis is particularly valuable when working with the subject of gender, because opinions about gender are so susceptible to ideology (Popper, 1971). CSCL researchers in general are vulnerable, as we were, to forming impressions based on the behavior of their most active users. Quantitative analysis is a useful partner to qualitative for understanding CSCL systems.

CONCLUSION

We initially designed the MOOSE Crossing environment with the goal of encouraging girls to become interested in technology. Evidence suggests that we have been partly successful in that endeavor. Mouse[1] (girl, age 9) says that she hates math, but loves to write programs on MOOSE Crossing. The following interview took place during an after-school program:

Amy: What's your favorite subject in school?
Mouse: Writing.
Amy: What kinds of things do you like to write in school?
Mouse: Stories about imaginary people.
Amy: Have you done any writing on MOOSE Crossing?
Mouse: Yes.

---

[1] All real and screen names of research subjects have been changed to protect their privacy.

Amy: What kinds of things do you write on MOOSE Crossing
Mouse: Programs, and....
Amy: How is writing a program different from writing a story?
Mouse: Programming it everything has to be right, so the thing you're making can work. But in stories it doesn't have to be really perfect-- It doesn't have to be so every word is correct.
[…]
Amy: What do you want to be when you grow up?
Mouse: I don't know
Amy: What do you NOT want to be when you grow up?
Mouse: I do NOT want to be... a mathematician!
Amy: How come?
Mouse: Cause I hate math?
Amy: How come you hate math?
Mouse: Cause... it's hard
[…]
Amy: "How come math is hard?
Mouse: I don't know…. If you're a mathematician you have to figure out hard problems.
Amy: But isn't figuring out a hard problem fun?
Mouse: No. It takes forever.
Amy: Is writing programs like doing math problems?
Mouse: No.
Amy: How come?
Mouse: Cause, it's *writing*, not working out problems! And you don't have to use the plus and minus.... and the equals, and the divide.
Amy: Now wait a second! You were just using a greater-than in your program. That's a math symbol!
Mouse: That's not a plus, a minus, a times, a divide, or an equals!
Amy: <laughs>
Mouse: It doesn't count
Amy: It doesn't count, OK.
Mouse: Go talk to somebody else!
Amy: Oh... OK….
Mouse: I'm working on something interesting!

While Mouse sees math as something she has no talent for or interest in, she is proud of her writing ability. In this environment, she sees programming as a form of writing. For at least this one child (and presumably some others), this environment has made computer programming more appealing than it likely otherwise would be. But if differences in achievement persist even in this environment, what is their source? Data analysis presented in this paper suggest that educators wishing to increase girls' level of technical achievement should explore strategies for increasing girls' interest in technical subjects. In both the BASIC programming environment Marcia Linn studied in the early to mid-1980s and in the CSCL environment we designed and studied from the mid 1990s to the present, girls program equally well as boys when they devote equal time to the activity.

ACKNOWLEDGMENTS

The authors would like to thank everyone who has contributed to the MOOSE Crossing project over the years, especially Mitchel Resnick, Elizabeth Edwards, Will Scott, Adam Skwersky, Trevor Stricker, and Steve Tamm. Current MOOSE administrators include Alisa Bandlow and Tysen Perszyk. Jason Elliott and Jon Orwant provided help with Perl scripts. Christian Jensen assisted with statistics. Special thanks to the kids, teachers, and parents who have participated in MOOSE Crossing. Research in the Electronic Learning Communities Group (http://www.cc.gatech.edu/elc/) at Georgia Tech is supported by IBM, Intel, Microsoft, Neometron, Ricoh, and the National Science Foundation.

REFERENCES

Abernathy, D. J. (2000). "How Men and Women Use the Internet Differently." *Career Journal*.

Bannert, M., & Arbinger, P., R. (1996) "Gender-Related Differences in Exposure to And Use of Computers: Results of a Survey of Secondary School Students" *European Journal of Psychology of Education 11*(3), 269-282.

Bruckman, A. (1997). *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids.* Unpublished PhD, MIT.

Bruckman, A. (2000). "Situated Support for Learning: Storm's Weekend with Rachael." *Journal of the Learning Sciences, 9*(3), 329-372.

Bruckman, A., & Edwards, E. (1999). *Should We Leverage Natural-Language Knowledge?* Paper presented at the CHI, Pittsburgh, PA.

Bruckman, A., Edwards, E., Elliott, J., & Jensen, C. (2000, June). *Uneven Achievement in a Constructionist Learning Environment.* Paper presented at the International Conference on the Learning Sciences, Ann Arbor, MI.

Colley, A., M., Gale, M., T., & Harris, T., A. (1994) "Effects of Gender Role Identity and Experience on Computer Attitude Components." *Journal of Educational Computing Research 10*(2), 129-137.

Giacquinta, J. B., Bauer. J., A., & Levin, J., E. (1993). "Beyond Technology's Promise." Cambridge England: Cambridge University Press.

Kersteen, Z., A., Linn, M., C., Clancy, M., & Hardyck, C. (1988) "Previous Experience And The Learning of Computer Programming: The Computer Helps Those Who Help Themselves." *Journal of Educational Computing Research 4*(3), 321-333.

Linn, M. C. (1985). "Fostering Equitable Consequences from Computer Learning Environments." *Sex Roles, 13*(3/4), 229-240.

Popper, K. (1971). *The Open Society and Its Enemies* (5th ed.). Princeton: Princeton University Press.