Chapter 58: Analysis of Log File Data to Understand Behavior and Learning in an Online Community

AMY BRUCKMAN
Georgia Institute of Technology

1. RESEARCH METHODS FOR STUDYING ONLINE COMMUNITIES

How do we study the behavior of users in online communities? Researchers from a variety of disciplines have evolved a rich set of both quantitative and qualitative approaches to studying human–computer interaction (HCI) and computer-mediated communication (CMC), and these methods are useful in the study of online communities. Unique to the study of CMC and online communities is the possibility of collecting *log file data*. It is possible for the computer to record every command typed by users—in some cases, every keystroke. In cases where users interact only online, we can collect a comprehensive record of all of their interactions. The completeness of the record and ease of collecting it are unprecedented.

However, log file data is more often collected than analyzed. We can save everything, but what does it mean? This chapter presents two examples of the use of log file data to understand user behavior and learning in one online environment, MOOSE Crossing. MOOSE Crossing is a text-based virtual reality environment (or "MUD") in which kids aged eight and older learn object-oriented programming and practice their creative writing. From analysis of log file data from the environment, we gained significant new insights into user behavior and learning. We will briefly discuss an example of qualitative log file analysis, in which a close reading of records of online interaction provides insights into how children learn from one another in this environment (Bruckman, 2000). The rest of the chapter contains an extended example of the use of quantitative log file analysis to try to understand whether there is any relationship between gender and programming achievement on this site. (An earlier version of this analysis was published in Proceedings of CSCL 2002 (Bruckman et al., 2002).)

Quantitative and qualitative methods are generally more powerful when used in a complementary fashion. In the data to follow, our qualitative findings suggested that girls are especially strong achievers in this learning environment. However, quantitative analysis of log file data shows that boys and girls use the environment in similar ways. A small group of girls form the core of "the regulars", the most active, dedicated members who are highly visible to both researchers and other members. The regulars have extraordinary levels

J. Weiss et al. (eds.), The International Handbook of Virtual Learning Environments, 1449–1465.

of achievement; however, their behavior is not typical. Among more typical members, achievement by boys and girls is effectively the same.

This points to a broader methodological issue in the study of online communities: *Researchers of online environments must be careful not to be over-influenced by the behavior of the regulars.* The most dedicated, regular members of an online community are the most visible. However, they are—by definition—not typical members. In this chapter, we present an example in which the behavior of the regulars is not representative of the behavior of the broader community population. This insight arose from the complementary use of qualitative participant-observation and quantitative log file analysis. This example highlights how using qualitative and quantitative methods together can be more powerful than either approach alone.

2. TYPES OF LOG FILE ANALYSIS

Jenny Preece notes that "the important first step before using data logging and metrics is to revisit the goals and the questions of the study" (Preece, 2000). A variety of approaches are possible, depending on the nature of the research questions. Log file analysis can be either qualitative or quantitative in nature. It can also be manual or automated.

Qualitative log file analysis is generally conducted manually (by a person interpreting logs); however, computerized tools can help the human reader. For example, software can help organize large amounts of data, and search for and identify excerpts that meet a desired set of criteria. The human reader in a qualitative analysis may use a number of theoretical frameworks to organize that analysis such as grounded theory (Glaser and Strauss, 1967), activity theory (Engestrom et al., 1999), distributed cognition (Hutchins, 1995), etc.

Log files need not and often should not be the sole source of data. Many studies of online communities pose questions about the meaning of a particular trace of behavior without taking the obvious step of asking the participants! Surveys and interviews with participants can greatly enhance a researcher's understanding of what is taking place in a log file. The converse is also true: A log file excerpt can be a jumping-off point to get strong interview data.

Quantitative log file analysis can be either manual (with a human translating log entries into specified metrics), or automated (with a computer program performing that translation). Some kinds of log file data lend themselves to automated analysis more readily than others. The nature of the research question affects whether manual or automated analysis is preferable. The scripting language Perl (http://www.perl.org/) has powerful pattern matching capabilities, and is often useful for automating work with log file data.

Quantitative analysis often simply measures amounts of activity over time. For example, Rick et al. analyze how much time students and instructors spend on a class CoWeb (collaborative website) to try to understand its cost effectiveness as a learning tool (Rick et al., 2002). Sometimes the absence of activity can be as revealing as its presence. Nonnecke and Preece look at postings to discussion lists to understand the lurkers (the more silent members) and what they contribute to the community (Nonnecke and Preece, 2000). Techniques from social network analysis can be used to show relationships among group members (Palonen and Hakkarainen, 2000). Techniques from discourse analysis can turn verbal data into quantifiable instances of phenomena of interest in a systematic fashion. For example, Susan Herring and colleagues use variations on this approach to understand diverse phenomena including gendered behavior online, power relationships in online discourse, online harassment, patterns of turn taking in online conversation, and emerging genres in weblogs (Herring, 2004b). Herring provides a useful overview of this technique in (Herring, 2004a).

Natural language understanding software may make it possible for automated analysis to ask questions about the content of discourse and interaction, not just the existence of activity. Activity on some sites follows regular patterns, and those regularities may be leveraged to allow easy automated parsing of activity. That is the case in the example that follows. The rest of this chapter presents a short example of qualitative and extended example of quantitative log file analysis of behavior and learning on MOOSE Crossing.

2.1. Example One: Qualitative Log File Analysis

To make a purely qualitative log file analysis possible, the log data and research questions posed about it must be intelligible to a human and interesting enough to hold their attention for extended periods of time. The paper "Situated Support for Learning: Storm's Weekend with Rachael" presents a qualitative close reading of log files from a holiday weekend when one girl (pseudonym "Rachael",[1] age 13) taught another ("Storm", age 12) how to program. Analysis of the record of their interactions was supplemented with face-to-face interviews and email exchanges. The resulting paper tells the story of what happened online during that weekend (Bruckman, 2000). Analysis of a single case provides one example of what can happen, but does not make any claims as to how typical the phenomena are.

Part of what made this approach feasible in this instance is the fact that the log files used are engaging to read, for both the researcher and the reader. The textual nature of the environment helps make this possible, because actions are summarized in succinct and easy-to-understand log file entries. In a MUD log, someone entering a room appears as a simple statement like "Amy climbs down from the branch above". In a graphical environment, such data may be recorded as a series of coordinate changes that are not intelligible to a human reader. In such situations, a program can be written to "play back" activity, using log file data to recreate an approximation of the original activity (at researcher-controllable speeds).

The multi-user nature of the environment also helps make qualitative analysis possible. In this example, as the girls explain what they are thinking to each other, they also explain their thinking to readers of the log. For example, at one juncture Storm repeatedly modifies a command on one of her projects, a virtual pet mouse. After many changes to the same line of code, we can infer that she is beginning to feel frustrated. However, we don't need to make that inference, because as soon as she meets up with her mentor Rachael, this exchange takes place:

Rachael says, "hi"
Storm says, "Hi!"
Rachael says, "Whatcha doin'?"
Storm says, "Being unsuccessful with that dratted mouse!"

A follow-up interview later confirmed that Storm was indeed quite frustrated at this point in time, and Rachael's sympathy and advice proved helpful in her progress on her project.

Qualitative and quantitative, automated and manual approaches to analysis of log file data are not mutually exclusive. The next, longer case study provides an example of quantitative log file analysis used together with qualitative participant-observation.

2.2. Example Two: Quantitative Analysis of Gender and Achievement

The MOOSE Crossing began with a set of questions about gender and computing.[2] 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 intervening years, much has changed about computers and computing technology. However, the basic facts of gender and computing for kids have not changed: Our results replicate Linn's early findings. In this case study, we observe student programmers in an environment that supports a mix of school and free-time home use. While boys develop significantly more programming expertise than girls (25% difference, $p = 0.004$), 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 choose to program both before and during their exposure to our programming environment, and time on task predicts level of achievement.

3. THE STUDY SITE: 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 and 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 et al., 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.[3] 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 wrote a series of Perl scripts to break down children's commands typed on MOOSE Crossing into categories (see Table 1).

Table 1. Categories of activity

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

Commands executed on MOOSE Crossing follow a specific syntax. Some commands are executed directly by the user, while others by the windowing interface, but all are recorded in the log and relatively easy to categorize. In many cases, the first word of a command identifies the type of command. For example, the command "create" makes a new object. Some cases are slightly more complicated. For example, there are an unlimited, user-definable number of commands that move a user from one room to another, but they have one easily detectable thing in common: The user's location is different after the command is executed. Online activity in this computerized environment is regular enough to make it easy to categorize, and much easier to work with than free-form English.

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 are not a lack of results—it in fact is 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.

4. DATA ANALYSIS: CATEGORIZING ACTIVITY

Through November 2000, 457 children 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 threefold increase in experimental subjects has led to a nearly fivefold increase in the amount of log file data. At the time of this analysis in November 2000, we had 3.4 Gb of data (compared to 0.7 Gb in 1997).

Table 2. Time on task (commands typed) by gender

	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

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-1990s, men and women were roughly equally represented online by the turn of the century. Our data on gender of participants is from a survey all members complete before joining MOOSE Crossing. This survey data is extremely helpful in interpreting the log file data.

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 are 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.

All transactions between client and server also recorded, allowing us, for example, 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).

One gender difference in this chart is statistically significant: Girls spend more time (as a percentage of total commands typed) communicating with others than boys (marginal significance: $p = 0.082$). This trend was also observed in the 1997 data analysis, but at the time was not 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

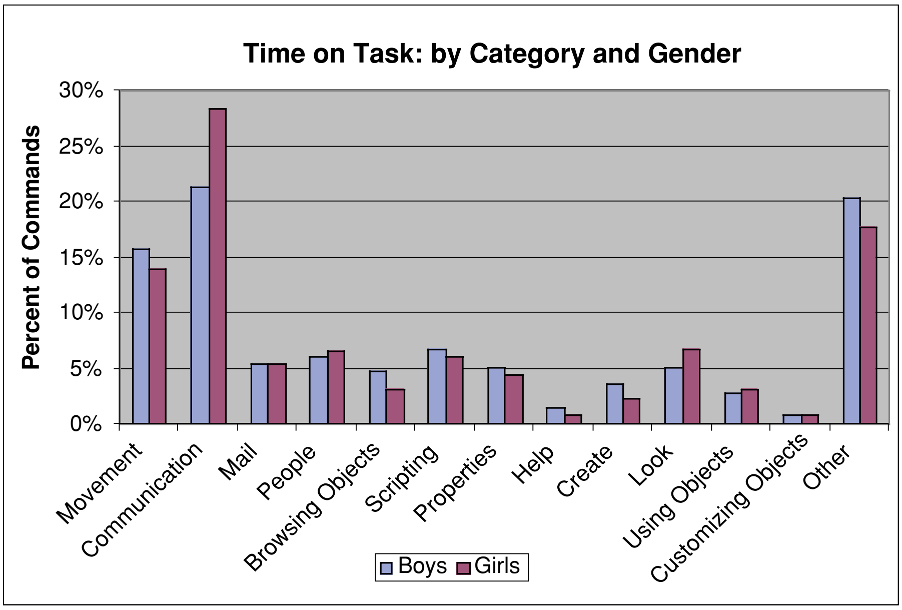

Figure 1. Time on task.

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.

5. 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 1–100.

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 have not.) Both new and old scales have the limitation that there are relatively subtle differences between

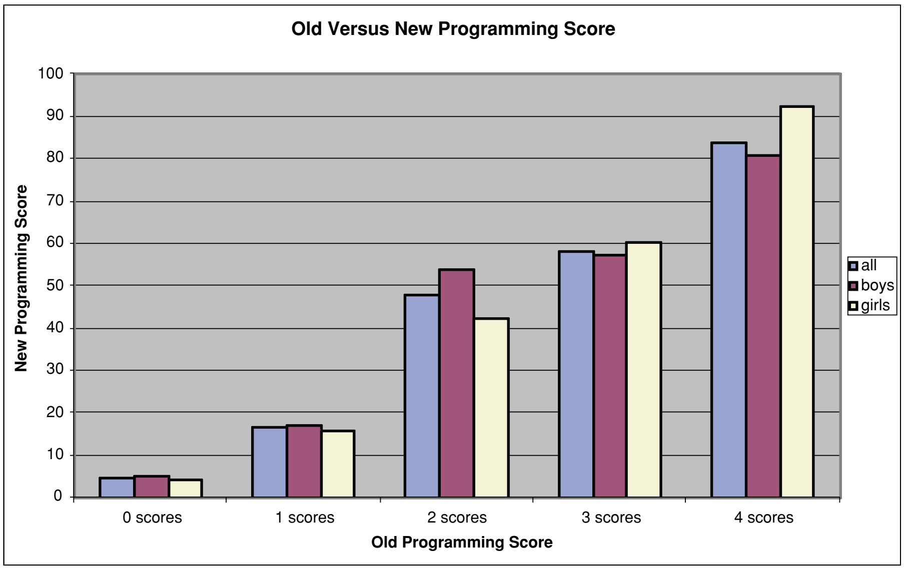

Figure 2. Old and new ways of measuring student programming achievement.

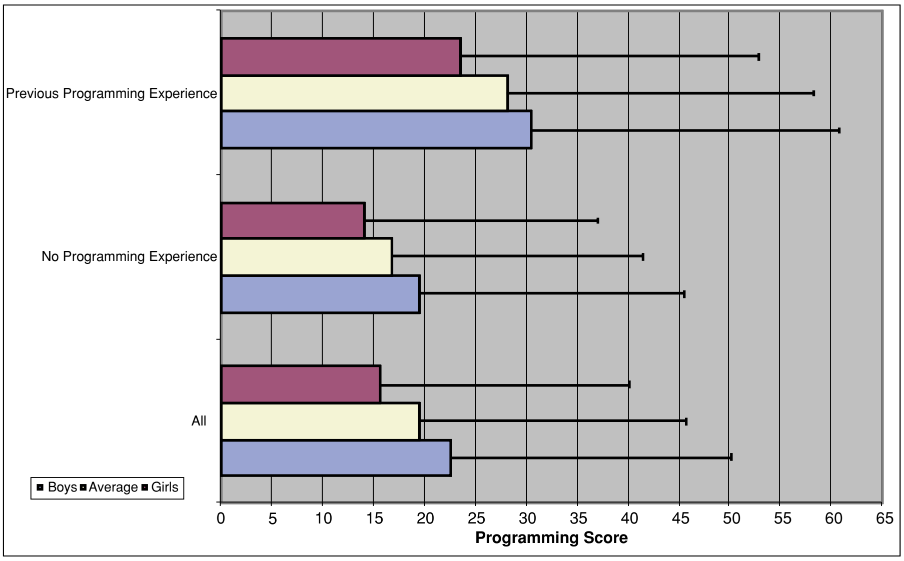

Figure 3. Prior programming experience and programming score.

categories two and three. One advantage of the new automated technique is that we can analyze all study participants, instead of a sample.

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 ($p = 0.001$) (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.

On first glance, boys have a higher level of programming achievement than girls ($p = 0.004$). However, regression analysis shoes 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 which of these are important and which are not. Regression analysis seeks to take a set of data-points and find the mathematical equation which best and most reliably models the data set given. 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 = 5.746
+3.645 (if the subject has previous programming experience)
−0.007 * (time talking to others)
+2.689e–07 * (time talking to others)
+0.006 * (time on task)
−9.616e–08 * (time on task)
+0.014 * (time spent scripting)
−6.950e–07 * (time spent scripting^2)

In other words, programming scores were positively related to time on task, time spent programming (effort), and previous programming experience. Programming scores were negatively related to the time they spent talking to others, or engaging in any activity other than programming for that matter (time on task is a zero-sum game). Gender, environment of use (home, school, or other) proved not to be statistically significant. (Adjusted R^2 = 0.76, S.E. = 12.8).

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 a more complex equation:

Time spent scripting = 66.42
+48.19 (if the subject is a boy)
+0.141 * (time talking to others)
−5.153e–06 * (time talking to others)
+3.33 * (help commands)
−0.003 * (help^2)

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 in the fact that communicating with others is a strong indicator for time spent on scripting. We also see that consulting the built-in help system has a positive effect. All other factors proved to be statistically insignificant. (Adjusted R^2 = 0.61, S.E. = 233.57)

6. 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 is not

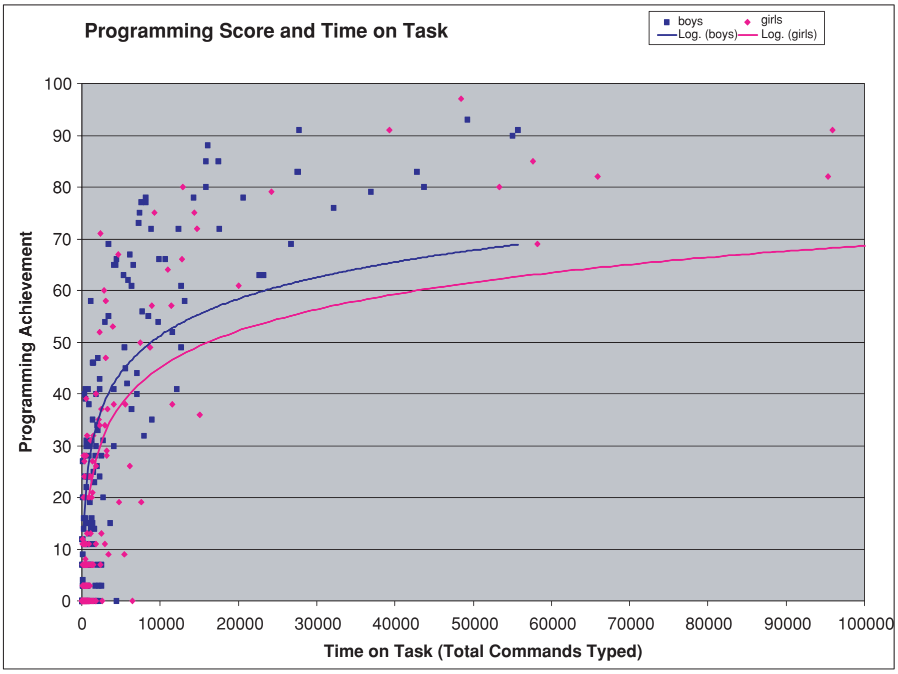

Figure 4. *Programming score and time on task.*

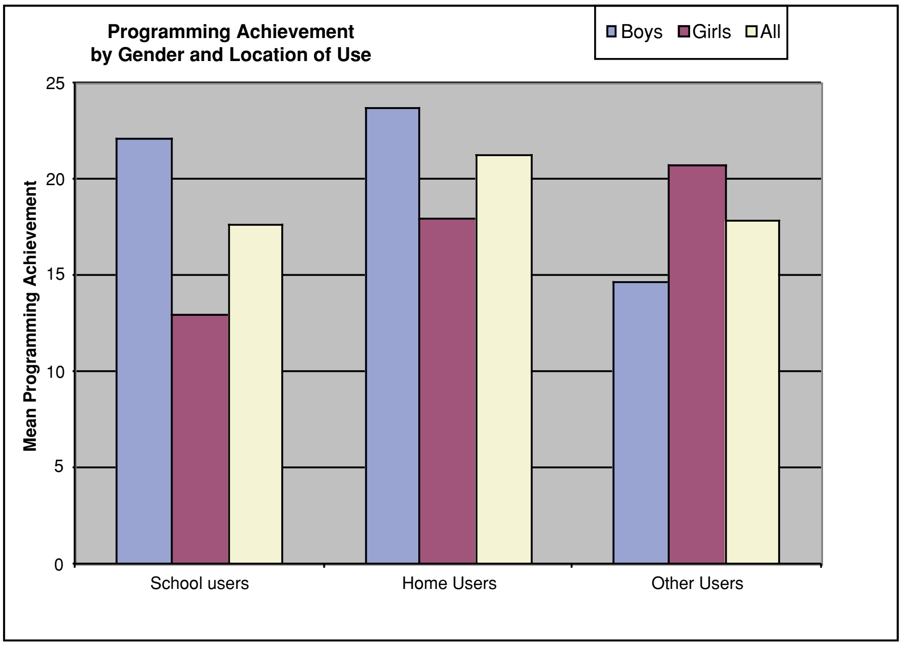

Figure 5. *Programming achievement by gender and location of use.*

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 1.54 more (6.5% increase) than those working from school; girls working from home score 4.98 more (35% increase) than girls working from school (Figure 5). (These figures are also suggestive but not significant.)

This apparent trend is consistent with our other findings. Girls tend on the whole 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.

7. COMPARISON OF QUANTITATIVE FINDINGS AND INFORMAL OBSERVATIONS

Since 1995, roughly a dozen administrators have spent each hundreds of hours 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. Girls like Storm and Rachael make a strong impression on observers. 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.

8. DISCUSSION

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 (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.

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.

9. CONCLUSION

A good log file is a terrible thing to waste. Use of log file data as a research tool can benefit from:

- Complementary use of qualitative and quantitative methods
- Both manual and automated methods of analysis
- Strategic leveraging of regularities found in the data from a particular domain
- Recognition of how "human readable" a particular data set is, and how this shapes the kinds of analysis possible
- Use of computerized tools to aid the human reader
- Complementary use of data from other sources (such as interviews, field notes, and surveys)
- Attention to ethical issues concerning the recording and analysis of log file data, maintaining respect for individuals' privacy and their rights as human subjects (in situations in which the work constitutes "human subjects research")

Effectively leveraging these strategies, log files can provide a wealth of information about online communities and the learning taking place there.

ACKNOWLEDGMENTS

Carlos Jensen and Austina DeBonte contributed to earlier versions of this chapter. The authors would like to thank everyone who has contributed to the MOOSE Crossing project over the years, especially Mitchel Resnick, Alisa Bandlow, Elizabeth Edwards, Naureen Hooda, Justin Jackson, Albert Lin, Tysen Perszyk, Will Scott, Adam Skwersky, Trevor Stricker, Steven Tamm, and Joseph Uhl. 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, Ricoh, and the National Science Foundation.

ENDNOTES

1. All names and online pseudonyms have been changed.
2. A review of the literature on gender and computing is beyond the scope of this chapter; see Margolis and Fisher (2001) for a good introduction to the subject.
3. Ethical issues in researching online communities are complex. Is it necessary to request informed consent from participants before recording online activity? If consent is generally required, is a waiver of the requirement for obtaining consent ever appropriate? Is consent required for analyzing records of online behavior already publicly available online? In what situations is research on data recorded online "human subjects research"? Each study must be dealt with on an individual, case-by-case basis. For a discussion of these issues, see Hudson and Bruckman (2004), Bruckman (2002), Kraut and Olson, et al. (2004).
4. All real and screen names of research subjects have been changed to protect their privacy.

REFERENCES

Bruckman, A. (1997). MOOSE crossing: construction, community, and learning in a networked virtual world for kids. Ph.D. Dissertation, MIT.

Bruckman, A., Edwards, E., Elliott, J., and Jensen, C. (2000). Situated support for learning: storm's weekend with Rachael. *Journal of the Learning Sciences* 9(3), 329–372.

Bruckman, A., Jensen, C., and DeBonte, A. (2002). Studying the amateur artist: a perspective on disguising data collected in human subjects research on the internet. *Ethics and Information Technology* 4(3), 217–231.

Bruckman, A. & Edwards, E. (1999). *Should we Leverage Natural-Language Knowledge?* CHI. Pittsburgh: CHI, PA: ACM Press.

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

Bruckman, A. & Jensen, C., and DeBonte, A. (2002). *Gender and Programming Achievement in a CSCL Environment*. Boulder, CO: CSCL.

Engestrom, Y. & Miettinen, R., and Punmaki, R. (Eds.) (1999). *Perspectives on Activity Theory.* New York: Cambridge University Press.

Glaser, B. L. & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago: Aldine.

Herring, S. (2004a). Computer-mediated discourse analysis: an approach to researching online behavior. In: Gray, J. H. (Eds.) *Desiging for Virtual Communities in the Service of Learning*. New York: Cambridge University Press, 338–376.

Herring, S. (2004b). Susan Herring's Homepage. Avaible at: http://www.slis.indiana.edu/faculty/herring/

Hudson, J. M. & Bruckman, A. (2004). Go away: participant objections to being studied. *The Information Society* 20(2), 127–139.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA: MIT Press.

Kraut, R., Olson, J., Banaji, M., Bruckman, A., Cohen, J., and Cooper, M. (2004). Psychological research online: report of board of scientific affairs advisory group on the conduct of research on the internet. *American Psychologist* 59(4), 1–13.

Linn, M. C. (1985). Fostering equitable consequences from computer learning environments. *Sex Roles* 13(3/4), 229–240.

Margolis, J. & Fisher, A. (2001). *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA: MIT Press.

Nonnecke, B. & Preece, J. (2000). *Lurker Demographics: Counting the Silent*. CHI. The Hague, The Netherlands: ACM Press.

Palonen, T. & Hakkarainen, K. (2000). *Patterns of Interaction in Computer-Supported Learning: A Social Network Analysis. International Conference on the Learning Sciences*. Ann Arbor, Michigan: Lawrence Erlbaum.

Popper, K. (1971). *The Open Society and its Enemies*. Princeton: Princeton University Press.

Preece, J. (2000). *Online Communities: Designing Usability, Supporting Sociability*. New York: John Wiley & Sons.

Rick, J., Guzdial, M., Carroll, K., Holloway-Attaway, L., and Walker, B. (2002). *Collaborative Learning at Low Cost: CoWeb Use in English Composition*. Computer Support for Collaborative Learning, Boulder. CO, Lawrence Erlbaum.