An Activity Monitoring System to Support Classroom Research

Hangjin Zhang, Kevin Almeroth
Department of Computer Science
University of California
Santa Barbara, CA 93106
{hangjin, almeroth}@cs.ucsb.edu

Monica Bulger
Department of Education
University of California
Santa Barbara, CA 93106
mbulger@education.ucsb.edu

Abstract: More effort has focused on integrating instructional technologies into classrooms than has focused on *assessing* the impact of these technologies on teaching and learning performance. To evaluate whether computer technology is beneficial to the teaching and learning process requires that we first develop a system to understand what students in a classroom do when there are computers in front of them. We identify the requirements for such a system and then describe a prototype system that uses off-the-shelf spyware as its main component. We report findings from a pilot study we conducted in a computer-equipped composition classroom. Our results show that while the monitoring system captured the necessary data, the use of an off-the-shelf tool did not completely satisfy all of our system requirements. Additional customization work on keystroke recording and password protection is necessary to effectively monitor real-time student in-class activities.

1. Introduction

Classrooms equipped with computers and Internet access are becoming increasingly common. Smartboards, Personal Digital Assistants (PDAs), and networked computers are being used in lecture presentations [5], for note taking [3], and in stimulating collaboration [4]. New computer and network technologies introduced into the classroom are altering both teaching and learning experiences [1]. With the goal of creating advanced classroom environments, more educators use computer-equipped classrooms or assume that students will bring their own computing devices to class.

While much effort has focused on introducing computer technologies into classrooms, not enough effort has been spent on *assessing* the impact of having computers in classrooms [2] [7]. One important first question to ask is whether these technologies help instructors teach more effectively and whether the technology helps students in their learning process. An equally important second question is whether there are any detrimental impacts of so much technology use. There are certainly different approaches to evaluate the impacts and answer these questions. Traditional studies involve surveys, self-report questionnaires, classroom observation, and retention testing [6]. However, these studies are prone to subjectivity and do not measure the real-time cognitive interaction between students and computers during a classroom lecture. This fact motivates us to design a classroom monitoring system and apply it in real classes to evaluate the in-depth impacts of computer-equipped classrooms. We believe that the analysis of data collected by such a system will provide better insight into the true impacts of classroom technologies.

In this paper, we take the first step in assessing the impact of having computers in the classroom. We design a student activity monitoring system that can help educational researchers study the impact of having networked computers directly accessible to each student during class. Specifically, the monitoring system has been designed to help educational researchers observe the students' behavior and further study student engagement in a lecture. In designing our monitoring system, we first identify the requirements of the monitoring system, then propose a general architecture and present a prototype implementation based on an off-the-shelf spyware program. Finally, we report our experience using this prototype system in a freshman composition course. The purpose of this case study is to understand: (1) whether the system is useful and how it can be improved, and (2) whether off-the-shelf software is sufficient, and if not, what additional features are necessary.

Our paper is organized as follows: Section 2 describes the system requirements for a classroom monitoring system; Section 3 describes the design of the architecture; Section 4 reports the results of our pilot study; and Section 5 discusses our conclusions and future work.

2. System Requirements

Most traditional studies of students' classroom activities involve surveys, self-report questionnaires, observations of classroom activities, and retention testing [6]. Our research requires, however, that we concretely measure the interaction between students and computers during a classroom lecture. To meet our specific needs, we require a system that can monitor student activity on each computer and associate it with observation and time-stamped video. This system must track student computer use during a lecture period and generate a comprehensive record of all student computer activity. This data must be rich enough for objective analysis, but require no human interaction during the collection period. In addition, to allow automated processing and analysis, the collected data must be organized into a computer-readable format, rather than an analog video clip.

Additional requirements of our system first include the collection of *sufficient* and *useful* data. A major concern in the collection of this data is the *privacy* of the student. Even within a research study, some data, such as account passwords, should be obscured to protect student privacy. Finally, our system should be easy to deploy and work efficiently even if there are a large number of computers in the classroom. We explore these requirements in the following sub-sections.

2.1 Sufficient Data

We consider the amount and type of data collected "sufficient" if the system records all relevant student activities and includes hints to gauge the significance of these recordings. We identified seven requirements for the kinds of data our monitoring system should collect:

1. **Application name.** The name of the application launched identifies the type of software and implies the kind of activity, for example Internet Explorer for web browsing, or Microsoft Word for document editing.
2. **Title of the active application window.** The title usually contains the name of the opened document or file, which is helpful in determining both where student attention was directed and whether the activity was relevant or not.
3. **Title of other visible application windows.** In addition to the "active window," it may also be helpful to record the "visible" windows. Visible windows are windows that were not active but were visible on the screen and may still have captured the student's attention, for example, a chess game.
4. **Timestamp when the application was launched.** A timestamp of when applications were started can be correlated with other data, for example, comparing one student's activity to another student, or with the instructor's lecture.
5. **Duration of application use.** How long an application was used can be correlated with other important events, for example, other active applications on the computer, instructor actions, or other student actions.
6. **Student/Computer interaction.** The interaction between the student and the computer, for example, keyboard and mouse activity, provides good insight into when and how the student is using the computer.
7. **Traffic traces.** Also important to collect is a record of the data packets exchanged over the network if the application is network-based. Network traffic information includes the destination address, the protocol or service, and the traffic contents.

In order to collect sufficient data, the monitoring system must monitor computers in real-time and continuously. An alternative would be to periodically *poll* computers and capture a screenshot of the current activity. While this solution would require less overhead and less storage, it would likely miss important activity that might occur between polling intervals. In addition, screenshots do not provide a complete log of student activities, but instead produce pictures that must be interpreted and analyzed at a later time.

2.2 Useful Data

While we advocate a relatively large set of specific data to collect, part of our decision process was to pay careful attention to not collecting *too much* data. A system that collects too much data could overwhelm a researcher with useless information and obscure the useful data. To ensure that we only collect data that is useful for our research purposes, we identify three guidelines for data collection:

- Only data that can be used to answer target questions should be collected. Collecting data that is not relevant to the specific research questions introduces overhead in collection, transmission, storage, and analysis. We must ensure that we collect minimally sufficient but maximally useful data. The one risk is that a researcher, in an attempt to be efficient, does not collect data that later becomes necessary. Clearly, there is a tradeoff between too much data and too little data.
- The data format should be both computer-readable and user-friendly to facilitate automatic and human analysis. Since there is likely to be a significant amount of data, the ease with which the data can be read is likely to be a major factor in how effectively it can be analyzed.
- In some cases, collecting sensitive information is prohibited in a computer lab or classroom; therefore, our system must not collect private or personal information.

2.3 Privacy Considerations

Privacy is a common concern for Internet users. It is of particular concern for us. Students who cannot trust an activity monitoring system to safeguard their privacy and anonymity will likely change their behavior. Even if students are unaware that they are being monitored, ethical behavior requires that we maintain student privacy and anonymity. Protecting students, therefore, is a key requirement for our system.

In a classroom environment, we are only interested in whether student activity is course-related or not. Identifying information, such as user logins or passwords is not required. For example, during a writing class, a student might log into eBay and submit a bid. We only need to record that the student visited a web site (and possibly the URL of the web site) at a particular time and for a particular amount of time. In this example, we are clearly not interested in the transaction itself. However, because in another example we might be interested in what the student does at a web site, e.g. how the student searches the Internet for information relevant to an assignment, we need to record some but not all information. For example, we should record students' keystrokes but not passwords or credit card numb ers. The monitoring system must be very thorough in what it collects, yet also very careful.

Given that part of prototype is based on available monitoring software, particularly spyware applications, we have had to pay special attention to build a system that only collects the data we need while respecting the privacy of our subjects. It is often the case that spyware aggressively tries to collect sensitive data. Therefore, we need to customize existing tools before we can directly use them in our monitoring system.

2.4 Scalability and Deployment

A classroom or lab may vary in size from five to fifty computers. The monitoring system must be able to accommodate all class sizes, even the largest. If class size is large, factors such as CPU load, network congestion, and storage limits may affect system performance. Our two main concerns are performance degradation and deployment overhead in large classrooms. Therefore, we identified the following four scalability requirements that guided the design of our system:

- Distributed agents should be used so that each agent is responsible for only one student computer and CPU load is distributed across multiple computers. Then the system can extend to a large class without worrying about the CPU load on any one computer.
- The monitoring system should use the network at off-peak times, rather than during class, to minimize the impact of large log file transfer.
- To handle the large volume of collected data, our system should also have sufficient data-handling capabilities that allow the data to be stored, retrieved, and analyzed efficiently.
- To avoid repeated configuration work on each computer, distributed agents should report their collected data to a central data center. Each monitoring agent should read a configuration file from a central directory and adjust its settings accordingly.

3. Architecture

After observing eleven composition lectures held in a computer lab over a twenty-week period, we determined that a system that included spyware as a monitoring agent could serve as a satisfactorily first step to record student activities both on and off the computer. A question we were only able to answer after a pilot study was whether the spyware had any limitations for such a use.

Spyware is a computer application that secretly monitors and collects user information or activities on a single computer and reports this information to a specified user. Spyware is often considered malicious because the program is installed without users' knowledge, and reveals personal information without their consent. In this study, we take advantage of spyware's capability to record and report user activities, such as keystrokes, mouse movement, application open/close events, and window information, but inform the student users that their actions are being recorded. We chose to use spyware instead of designing and writing our own software because of the time saved by using an existing application. We could have easily taken more time and written our own code, but we were hopeful we could more quickly develop a functional system.

After a thorough functionality and features comparison, we chose to focus our attention on two spyware programs: Boss Everyware (http://www.bosseveryware.com/) and SpyAgent (http://www.spytech-web.com/). Both programs allow users to monitor more than one client and display a rich set of information such as the applications a user has executed, actions performed within the application, and duration of application use. In addition, both programs have a reasonably sophisticated Report Manager. The primary difference between these two tools is the way they deal with the keystroke and clipboard logs. Boss Everyware does not record clipboard logs at all, while the SpyAgent provides a chance to access and record the application clipboard. As for the keystroke log, both record every raw keystroke, whether they are printable characters or function keys. Boss Everyware is superior to SpyAgent because it also provides a high-level key interpretation function, which is useful in recovering the typed text. For example, when the "backspace" key is pressed, SpyAgent only inserts the key code into the event sequence, whereas Boss Everyware is smart enough to understand that the stroke is a correction to the text previously entered and simply stores the corrected text. The log generated by Boss Everyware, therefore, is more readable and meaningful than the log generated by SpyAgent. Because of this difference, we started with Boss Everyware as the code framework for our monitoring agent.

Our prototype monitoring system has three input sources: the monitoring agents (one for each computer), the network traffic watcher, and the class context recorder. All collected information is gathered at a data center. After a pre-processing step, the data is then transferred to an analyzer and report generator, which creates a human-readable activity report. Figure 1 shows the high-level architecture for our monitoring system and how its components fit together.

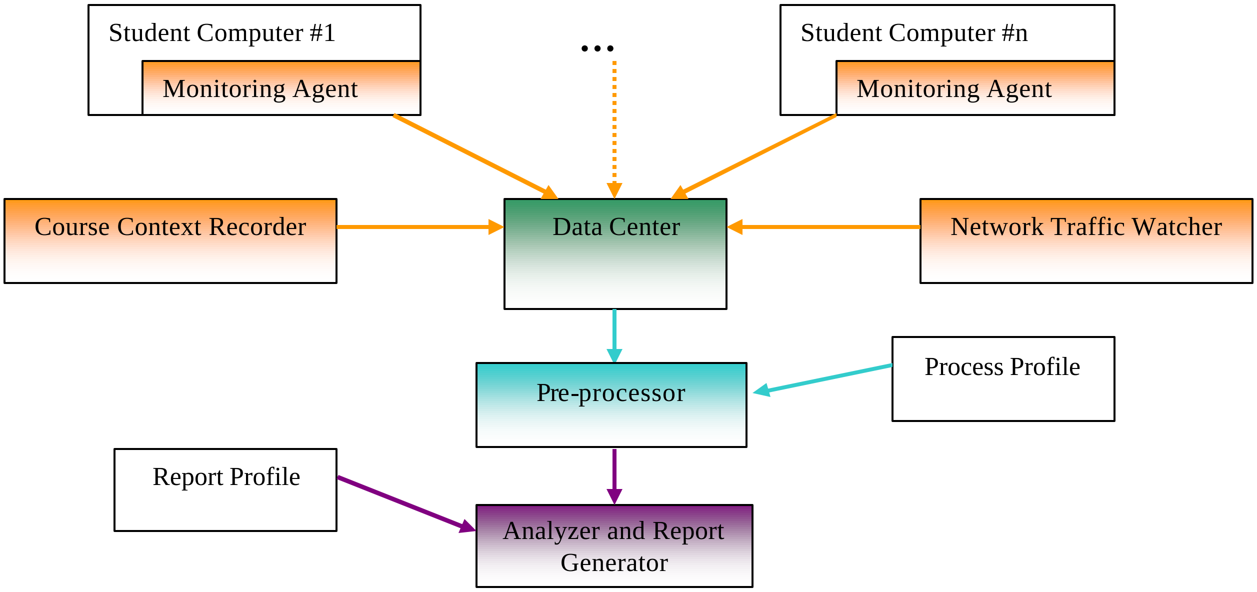

Figure 1. Monitoring system components.

As described above, there are three sources of data for the monitoring system. The first is the *monitoring agent*. The monitoring agent is the Boss Everywhere spyware we described earlier. A copy is installed on each student computer. We adopted a distributed monitoring architecture that uses multiple monitoring agents. The monitoring agents are responsible for collecting all computer-related activities, such as application information, window opening/closing, keystrokes and mouse movements. In our distributed architecture, each agent is responsible only for the machine on which it is running.

The second source of information is the *network traffic watcher,* which sits at the boundary of the classroom network and the rest of the Internet. It collects all network traffic exchanged between the classroom and the outside world. This information is a complement to the data collected by the monitoring agents. While the monitoring agents collect what the user does, the network traffic watcher collects the resulting data that flows across the network.

The third source of information is the *course context recorder*. Context information is recorded using both an observer writing field notes and a digital video camera with timestamp information. By looking at the context record at a given point in time, we are able to correlate physical activity with computer activity and identify behavior trends.

Our architecture design requires all logged data to be collected at a central repository. The *data center* component is introduced for this purpose. A data center is a sink point for all streams of collected information. For a small-size classroom, a personal computer can serve as the data center. When the size of the classroom increases, we can replace this component with a professional database on a high-performance workstation.

As the data is collected, it is pre-processed. This *pre-processor* component satisfies our useful data and privacy protection requirements. Because the monitoring agents and network traffic watcher may capture sensitive data such as passwords, the pre-processing component is responsible for filtering this information. The second purpose of the pre-processing component is to normalize the data format, making data analysis easier. Since the monitoring agents and network traffic watcher do not have the same data format, a pre-processing procedure is responsible for converting data from the two sources into a consistent format for later analysis.

The final component in the architecture is the *analyzer* and *report generator*. The function of the analyzer and report generator is to provide access to the database and create user-friendly reports for analysis. Because the researcher is likely to want the output in a specific format, we simply make the data available in a simple and consistent format.

4. Pilot Study

We used our prototype monitoring system in an on-campus lab equipped with 25 computers. Our metrics for evaluating the success of our prototype included both the system's ability to record significant student activities as well as its ease of deployment.

We observed a two-hour class by using time-stamped video, a human observer, and a limited version of our monitoring system. As mentioned above, the spyware monitoring agent recorded all application activities, including the time the application was opened and closed, the duration of time the application was used, periods of inactivity, and, in the case of Internet use, the URLs for each website and the duration of time spent at each address. The monitoring system was successfully deployed remotely on all of the computers using a single configuration. Unfortunately, the keystroke recorder had to be disabled because we could not be absolutely sure it would not collect some sensitive data. Using the data collected, we observed patterns of student computer use including sharp rises and decreases in Internet activities.

The log files, combined with the time-stamped video recording of the lecturer and observations of student activities, produced a rich data set. We used the log files to create a database of individual student profiles and mapped their Internet use. We also compiled an activity log of class Internet use, broken down by minute. We were able to observe patterns of classroom Internet use and to compare periods of high Internet activity with both the content of the lecture and in-class behavior to discover cues for engagement and disengagement.

Overall, our monitoring system proved beneficial for classroom research. It allowed us to "get inside the heads" of our students and see where they focused their activity during a class period.

Even though our monitoring system was useful, it had limitations that resulted in three main challenges. First, the system administrator required us to disable the keystroke recorder because the Boss Everyware software also captures passwords for some applications. We felt passwords were private and their collection violated student rights. The absence of the keystroke recorder limited our access to important data, such as text typed in online text fields and within Microsoft Word. The functionality of the keystroke recorder will be more carefully considered in future studies.

The second challenge emerged in the deployment stage. Because we had no direct access to the computer lab, we had to rely on the system administrator to install and uninstall the software. This introduced significant workload for the administrator. We were able to work around this difficulty to some extent by creating a single configuration for remote deployment on all of the lab computers.

A final challenge came in the analysis stage. Since spyware is produced primarily for businesses, it did not have the necessary research tools for organizing and analyzing the logs it produced. We thus had to analyze the data manually. With improved analysis tools and keystroke recording functionality, spyware will prove to be a more effective tool for educational research.

5. Conclusions and Future Work

Computer technologies and devices are increasingly being introduced into classrooms to enhance the learning experience. Assessing the impacts of these technologies is an important research objective. In this study, we designed a student monitoring system to record the cognitive interplay between students and their computers. Introducing this monitoring system is a step toward assessing the effectiveness of technology in the classroom.

In our study, we first identified the overall requirements that such a monitoring system must meet, and presented a general architecture as a guide for its implementation. Next, instead of building our system from scratch, we developed a prototype system using an off-the-shelf spyware program combined with a small number of supporting configuration and analysis scripts. This solution significantly reduced the software development burden.

In order to evaluate our prototype implementation, we conducted a pilot study in a freshman composition course. The case study first shows that the monitoring system is capable of capturing enough activity data to analyze student behavior. It also shows that using off-the-shelf tools does not satisfy all of our research requirements. Some substantial customization work, such as keystroke recording and privacy protection, is necessary to effectively monitor student activity in a classroom setting.

In this paper, we have focused on a *resource-controlled classroom* where computers are provided as part of the classroom infrastructure and therefore are controlled by the educational institution. Since a system administrator could install any application or program on these computers, our monitoring system is relatively easy to design and deploy. However, not all classrooms are resource-controlled. Future studies should examine ways to monitor student activities in an *open infrastructure classroom*. In an open infrastructure classroom, students bring their own laptops or PDAs to class and access an open network. While collecting network traffic may be possible, the system administrator cannot simply install monitoring software on students' personal computers. Assuming that students in an open infrastructure classroom cannot, or are not required, to install monitoring system on their computers, the monitoring problem becomes much more challenging. A possible solution would be to collect available network traffic and construct a representation of what students are doing with their computers. This solution requires fundamentally different analysis methods and we leave the problem for future work.

References

[1] Cuban, L. Oversold and Underused: Computers in the Classroom. *Harvard University Press*, Cambridge, 2001.

[2] Shotsberger, P. and Vetter, R. Teaching and Learning in the Wireless Classroom. *IEEE Computer*, Volume 34, Pages 110-111, March 2001.

[3] Singh, G., Denoue, L. and Das, A. Collaborative Note Taking. *Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education*, Pages 163 -167, March 2004.

[4] Anderson, R., VanDeGrift, T., Wolfman, S. and Yasuhara, K. Promoting Interaction in Large Classes with Computer-Mediated Feedback. *Proceedings of Computer Support for Collaborative Learning,* June 2003.

[5] Anderson, R., Simon, B., Wolfman, S., VanDeGrift, T. and Yasuhara, K. Experiences With a Tablet PC Based Lecture Presentation System in Computer Science Courses. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*, Pages 56-60, March 2004.

[6] Fredricks, A., Blumenfeld, C., and Paris, H. School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research*, Volume 74, Pages 59-109, 2004.

[7] Foster, K. Are Classroom Computers Worthwhile? *IEEE Spectrum*, Volume 39, Page 60, February 2002.