Moodog: Tracking Students' Online Learning Activities

Hangjin Zhang[1], Kevin Almeroth[1], Allan Knight[1], Monica Bulger[2], Richard Mayer[3]
[1]Department of Computer Science, [2]Gevirtz Graduate School of Education, [3]Department of Psychology
University of California
Santa Barbara, CA 93106
[1]{hangjin, almeroth, aknight}@cs.ucsb.edu, [2]mbulger@education.ucsb.edu, [3]mayer@psych.ucsb.edu

Abstract: Many universities are currently using a Course Management System (CMS) to organize course materials and conduct online learning activities. However, most commercial or open source CMS software does not include comprehensive access tracking and log analysis capabilities. In this paper, we propose and implement a CMS log analysis tool, called `Moodog`, to track students' online learning activities. The goals of `Moodog` are twofold: first, to provide instructors with insight about how students interact with online course materials, and second, to allow students to easily compare their own progress to others in the class. With the activity information available, `Moodog` also provides automatic email reminders to students encouraging or instructing them to view available materials that they have not yet accessed. Using `Moodog`, we have conducted a preliminary case study on a course held in Spring 2006 at the University of California, Santa Barbara to reveal activity patterns that occurred in this course.

Introduction

In recent years, many university instructors have started to use Course Management Systems (CMSes) to manage or distribute course-related materials and conduct online learning activities. Students access course materials via the CMS, as well as participate in other learning activities, such as submitting homework assignments and posting discussion messages in online forums. Preliminary studies indicate that CMSes have the potential to both increase interaction beyond the classroom (Knight, Almeroth & Bimber, 2006) and positively affect student engagement with the course materials (Harley, Henke & Maher, 2004).

However, most CMSes lack the comprehensive function to track, analyze, and report students' online learning activities within the CMS. As reported in a previous study (Hijon & Carlos, 2006), the built-in student tracking functionality is far from satisfactory. The common problem is that a CMS only provides low-level activity records, and lacks higher level aggregation capabilities. Without students' learning activity reports, an instructor has difficulty in effectively tailoring instruction plans to dynamically fit students' needs.

Our research focuses on exploiting available CMS logs to provide learning activity tracking. One possible way in which CMS log data can be used is to visualize learning activity data via a more human-friendly graphical interface. This interface is updated in real-time to reflect all activities, and is integrated into the original CMS. Using this graphical interface, instructors can gain an understanding about the activities of the whole class or can focus on a particular student. Students can use the interface to understand how their progress compares to the whole class. Our hypothesis is that the availability of this graphical interface can provide important insight into how students are accessing online course materials, and this insight will be useful to both the instructor and to students.

We first examine what is available in CMS logs, and identify the key metrics important for understanding students' use of online course materials. We then design a general architecture for extracting such meaningful metrics from the low level logs, and develop visualization techniques customized for the different student and instructor perspectives. Beyond extracting and visualizing the activity data, we also extend the system with the capability to automatically send reminder emails to students who have not accessed key course materials. Finally, we use our tool to analyze on a real course presented at the University of California, Santa Barbara (UCSB), and focus on an analysis of student activities recorded in the CMS logs.

Our log analysis tool has been developed for the Moodle CMS (Moodle, 2006), which is why we call our system the "Moodle Watchdog", or `Moodog`. `Moodog` is superior to the original Moodle log facility in several aspects: (1) it provides aggregated and meaningful statistical reports; (2) it visualizes the results, which makes comparisons between multiple students much easier; (3) not only does `Moodog` display the activities that a student has performed, but it also identifies the materials a student has *not* yet viewed; and (4) it has the capability to remind students to view those materials that they have not yet downloaded.

The remainder of this paper is organized as follows. First, we review related work. Next, we present the design and implementation of our `Moodog` system, followed by a case study that uses `Moodog` to analyze a course at UCSB. The paper is then concluded with a vision of future work.

Related Work

Utilizing logs to track user activities is not new, and is used in a wide variety of computer applications. For example, almost all online stores collect and analyze customers' purchase history for better customer care or advertising (Sen, Dacin & Pattichis, 2006). Many commercial web sites mine customer interests and trends from web server logs, and aim to provide more tailored services. However, this level of log analysis is not widely implemented in CMS software. In a related study (Hijon & Carlos, 2006), the authors compare the tracking functionality of various CMS tools. Their conclusion is that none of them offer much tracking ability despite the strong desire for such functionality.

Among a few similar projects, the one most relevant to ours is `CourseVis` (Mazza & Dimitrova, 2004), which utilizes information visualization and graphical representation techniques to display web log data generated by the WebCT CMS. The main goal of CourseVis is to help teachers become aware of the social behavior and cognitive aspects of remote learners. There are two main differences between `CourseVis` and `Moodog`: (1) `CourseVis` is a stand-alone application, while `Moodog` is integrated into Moodle as a plugin; and (2) `CourseVis` only visualizes students' online activities, while `Moodog` is also able to send automatic reminder emails to students.

Visualization is another important related topic. The `Pulse` project (Chen, 2003) uses a simple system to visualize the "pulse" of a classroom. The idea is to offer a clear overview of remote classroom activity at a glance. The authors build a multi-party video-conferencing system that automatically determines students' in-class activity, such as speaking, making gestures, or moving between seats. The different activities are annotated with different colors in the video as an activity indicator. The activity indicator reflects the classroom interaction dynamics, and acts as a measure of the pulse of the classroom. The difference between `Pulse` and `Moodog` is that the former visualizes the physical activities or body movements of students, while `Moodog` tracks students' online activities.

In e-learning more generally, personalized and adaptive learning support is more important than in traditional classrooms. `Moodog` provides simple and basic functionality for personalized learning assistance by sending automatic reminder emails. In one project (Dolog, Henze & Nejdl, 2004), the authors propose a complicated service-based architecture to establish personalized e-learning in a distributed learning environment. One of the important steps is to extract the learner's profile. We believe `Moodog` provides a possible means to build this profile. Similarly, in another project (Clayden & Warren, 2006), the authors investigate an adaptive e-learning service based on learning styles. `Moodog` could again prove useful to such a project by helping to identify student learning styles.

Moodog Design and Implementation

Activity Related Statistics Variables

A single Moodle log record consists of four dimensions of information: *who*, *what*, *when* and *where*. Compared to a single log record, we are more interested in the properties exhibited by a group of records. For example, we want to be able to answer questions, such as, for Student A, how many resources has she downloaded or viewed over the entire quarter? Often, we want to limit the analysis on one or two of the dimensions, and query a property of the other. For example, the property could be the record count, minimum, maximum, or average value of the available records. Our goal is to design a query system that attempts to answer a fixed set of pre-defined, yet commonly needed criteria and questions.

Because the queries are fixed, it is therefore important to determine which sets of data are useful enough to be included in `Moodog`. We categorize all statistics into four groups: data about the course, data about the students, data about the resources, and data about access times. A brief description for each of these categories is as follows:

- *Course summary*. This view of statistics lists overall information about a particular course, such as how many students are enrolled; how many resources are available to view; how many times (sessions) each student has logged in; and the most active student and the most popular course resource.

- *Per-student statistics*. This information allows an instructor to take a closer look at a particular student or make a comparison between multiple students. This set of information includes: how many times a student has viewed each course material; how many sessions each student has; how much time a student spends on CMS; how many resources the student has and has not accessed; and forum activity, including how many discussion threads initialized by the student and how many follow-up messages the student has posted.

- *Per-resource statistics*. This view is from a resource perspective and allows the instructor or students to check which materials have been accessed and which not. For each resource, this view lists how many unique students have accessed the material, and how many total times it has been accessed. It also lists those students who have never viewed this particular resource.

- *Time-based statistics*. This information allows an instructor or students to identify the times when people are interacting with the Moodle site. The view offers three different time periods: week, day of the week, and hour of the day.

Interface Design

After aggregating the desired activity data, we present the results by integrating `Moodog` reports into Moodle, instead of making a separated application. In this way, users need to be authenticated only once, thereby eliminating the need to maintain two sets of user accounts. At the same time, we try to maintain the original Moodle interface as much as possible. Our reasoning is that the less we clutter the interface, the better. As a result, we put the detailed results in separate pages and insert only a few links in the main Moodle page. Students can easily navigate to the detailed `Moodog` results without being overly distracted by the newly added features.

In order to make the activity report intuitive and straightforward, we rely heavily on graphics to represent `Moodog` data. Figure 1 shows the modified main page of Moodle. The highlighted ovals represent our interface changes from the original Moodle interface. The most important interface change is that we add a green-and-red bar to the right side of each course resource. The bar is a popularity indicator for the corresponding resource and consists of two segments: the green segment representing the percentage of students who have viewed the resource, and the red segment represents the percentage of those who have not viewed the resource. Using this bar, students and the instructor can get a quick sense of how many students have or have not viewed a particular resource.

Because of the different roles and permissions of an instructor versus a student, we have designed slightly different interfaces for each. The main differences are (1) in the student's interface, the identifying information of the other students is removed; and (2) in the detailed resource view, the instructor can see the list of students who have not visited this resource, and have an option to send a reminder email to those students.
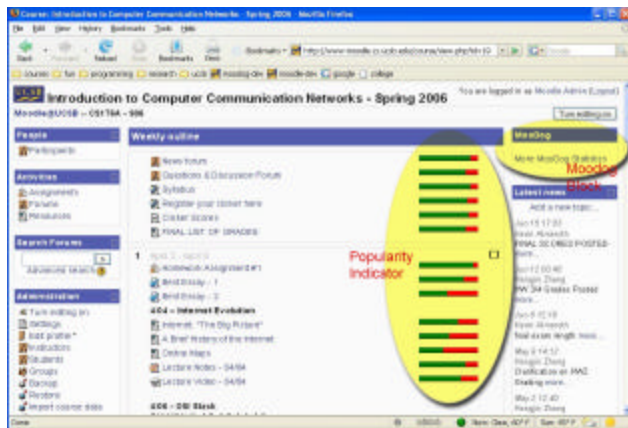

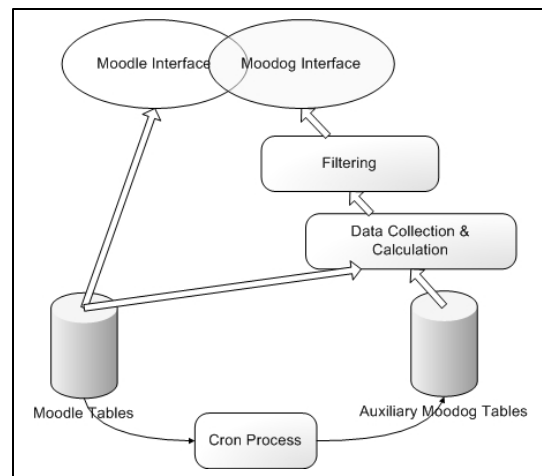Figure 1: The Moodle main page with `Moodog` bars.

Figure 2: The `Moodog` architecture.

Feedback and Reminders

In addition to collecting and presenting student activity data, we can proactively provide feedback to students or the instructor. `Moodog` tracks the Moodle logs, and when certain conditions are met, `Moodog` automatically sends an email to students to remind them to download or view a resource. For example, if a student has not viewed a

resource before a set time, the system sends an email. Another example is to send a reminder depending on the percentage of students who have viewed the resource. If more than a certain percentage of students have viewed the resource, the system triggers the email to those who have not done so. In addition to these two automatic reminders, an instructor can manually trigger emails to be sent by clicking on a button in the web page. For the email to be sent, the instructor can edit the message he/she wants to send.

Implementation

Figure 2 shows a representation of the `Moodog` architecture. `Moodog` consists of four main parts: a web interface, filtering facility, data collection & calculation, and the "cron" process components.

The `Moodog` interface is responsible for formatting activity results, both for the instructor view and the student view. The filtering component is used to remove unwanted results. For example, if a student has already dropped the class but has not un-enrolled from the Moodle site, the date collection component still counts the student in the report. The filtering component provides a mechanism to allow an instructor to filter unwanted results.

The two remaining components are the data collection & calculation component and the cron process. The data collection & calculation component reads the Moodle logs and other course related information, such as username and course meta-data from the Moodle tables, and reads other `Moodog`-related data from the auxiliary `Moodog` tables. Some activity-related data, such as session information, cannot be retrieved from the database tables by a single SQL statement. Rather, it requires a series of queries and takes significant time to complete the necessary calculations. In order to improve the interface response time, `Moodog` runs the script offline, and saves the results to an auxiliary `Moodog` table. These scripts are run by the cron process component once an hour.

Case Study

Although the initial goal of `Moodog` is to let the students and the instructor see students' activity and progress during a course, we can also use `Moodog` to calculate statistics after a class has ended. In this section, we use `Moodog` to analyze a course held in Spring 2006 at UCSB. The course is CS176A, entitled, "Introduction to Computer Communication Networks", which is an upper division course for senior students in the Computer Science major. There were 38 students enrolled in the course, each of which had registered with the Moodle site. All students agreed to participate in the study and have their Moodle activities recorded and analyzed.

Per-User Activity

`Moodog` provides a rich set of information about how often and how intensely students interacted with Moodle. Figure 3 shows the student activity statistics for the CS176A course. The statistics show six fields of data: total number of views, total number of sessions, total online time, number of viewed resources, number of initial threads posted by the user, and the number of follow-up messages posted. Normally student names are provided in the instructor view, however, to protect student privacy, student names have been removed from Figure 3.

The first line of the table shows the average value for all students for each variable. The average view count is 357, and the maximum is 776. The average number of sessions is 60, and the maximum is 130. And the average online time is 7 hours 15 minutes, while the maximum online time is 16 hours 32 minutes. In terms of the number of visited resources, we note that only two students had viewed all 52 resources posted on the course page. The average number of viewed resources is 36.

Our preliminary analysis shows that there is a positive correlation between a student's forum activity and the overall course grade. Students who frequently viewed the forums performed better on course assignments and exams. In future work, we plan to identify which online activities predict academic performance.

Resource Coverage

In addition to a view from a per-user perspective, we can also view the activity logs from a resource perspective. Figure 4 shows the number of students who viewed each resource. For each resource, the table shows the number of unique students that viewed it, and the number of total view counts by all students. From the figure, we can conclude that homework assignments are visited more often than any other resource. The top 5 popular resources are HW#2, HW#4, HW#1, News Forum, and HW#3, respectively. Because Figure 4 is truncated, the instances of these assignments are not shown in this figure.

The next popular category of resources is lecture notes. On average, every set of lecture notes is downloaded or viewed by about 30 students. In this course, we also made lecture videos available. We found that every video is visited

by at least 14 users. On average every video is viewed by 18 students, which is half of all students. Although we did not thoroughly study the impact of video availability on learning performance, the count itself suggests that students are interested in viewing lecture videos.

For the entire set of 38 students, no resource was viewed by every single student. The most popular resources were only viewed by 36 of the students. Further examination showed that two students dropped the class in the first or second week, and two more students dropped around the middle of the quarter. None of these students, however, un-enrolled themselves from Moodle. Therefore Moodog lists them in the statistics results. As a result of this problem, we developed a filtering mechanism for the instructor to remove these invalid records from the final result.

From time to time in this course, the instructor updated a resource listing the overall ranking of students in the course. This resource was available via Moodle with the names removed (but based on the grades list, a student could identify which record was his or hers and therefore tell his or her rank in the class). We found students were interested in this updated grade report. The view count for this resource was greater than almost all of the lecture notes.
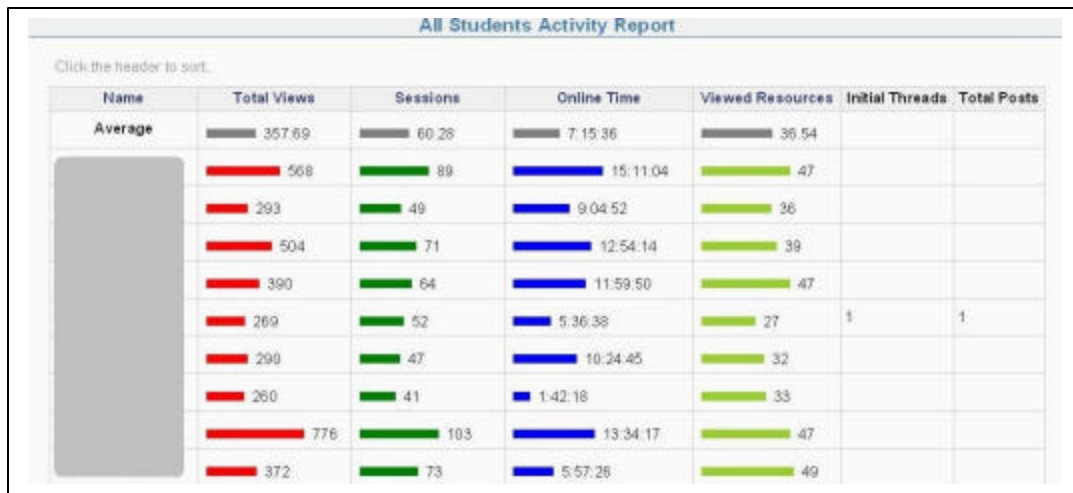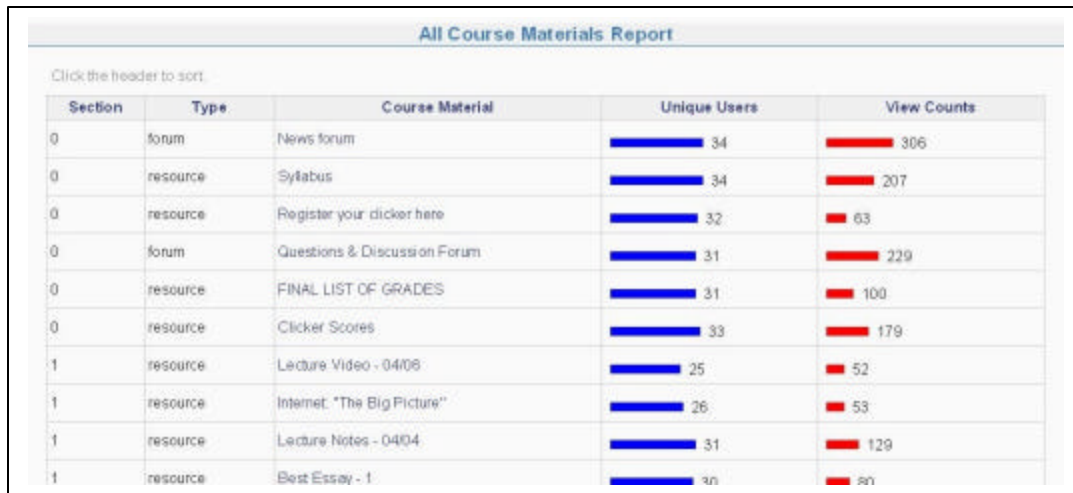

Figure 3: Per-user activity as measured by Moodog.

Figure 4: Resource access coverage as measured by Moodog.

Access Time Pattern

In addition to the session intensity and resource access coverage, we also examined the pattern of when resources were accessed. We used three different time granularities in our review: on a weekly basis, on a daily basis, and on an hourly basis. Due to pages limitation, the figures are not included.

From the weekly statistics, we find that Week 11 and Week 5 had the most activity. The reason is that there was a midterm in Week 5 and the final exam was in Week 11. Most students logged into the Moodle site to review the course materials the week of the exams. In terms of days of the week, Monday and Wednesday received slightly more visits than other days. This result occurred because the lectures are held on Monday and Wednesday and lecture notes were posted the morning before each lecture. For the hourly statistics, it is not surprising to find that early morning hours (3am - 7am) had significantly fewer accesses than at other times during the day.

Conclusions

In this paper we report on our effort to design and implement the `Moodog` application to track students' online learning activities based on CMS logs. We then visualize the results with a simple graphical interface. `Moodog` is also able to automatically send feedback and reminder emails based on students' activity histories.

We have integrated the `Moodog` module into the Moodle CMS and made some small interface changes to Moodle. Intuitively, the presence of the popularity bars should encourage students to check the course materials more frequently and promptly if he or she sees most of their classmates have already done so. Our hypothesis is that the availability of `Moodog` statistics will positively affect both how an instructor adapts the course and how students learn. However, we have not yet thoroughly evaluated the impact of using `Moodog`. In future work, we plan to validate our hypotheses.

Running `Moodog` on a past instance of a course reveals a significant number of interesting user behavior patterns. In this paper, although we only conducted an off-line analysis after the course was finished, the same analysis can be conducted in real-time during the period when the course is happening.

Monitoring student learning activity is an essential component of high quality education, and is one of the major predictors of effective teaching (Cotton, 1998). Research has shown that letting a learner know his or her ranking in the group results in more effective learning. `Moodog` provides a possible means for students and the instructor to receive this feedback. In future work, we plan to more comprehensively evaluate the impact of `Moodog` statistics on student academic performance.

References

Chen, M., "Visualizing the pulse of a classroom", in *Proceedings of the eleventh ACM international conference on Multimedia*. New York, NY, USA: ACM Press, November 2003, pp. 555-561.

Clayden, P. and Warren, I., "An investgation into adaptive e-learning based on learning styles", in *Proceedings of the 18th World Conference on Educational Multimedia, Hypermedia & Telecomuunications*, June 2006, pp. 2667-2672.

Cotton, K., "Monitoring student learning in the classroom", *in School Improvement Research Series Close-Up #4*, May 1988. [Online]. Available: http://www.nwrel.org/scpd/sirs/2/cu4.html.

Dolog, P., Henze, N., Nejdl, W., and Sintek, M. ,"Personalization in distributed e-learning environments", in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. New York, NY, USA: ACM Press, 2004, pp. 170-179.

Harley, D., Henke, J., and Maher, M.W., "Rethinking Space and Time: The Role of Internet Technology in a Large Lecture Course", in *Innovate*, vol. 1, no. 1, 2004. [Online]. Available: http://innovateonline.info/index.php?view=article&id=3.

Hijon R., and Carlos, R., "E-learning platforms analysis and development of students tracking functionality", in *Proceedings of the 18th World Conference on Educational Multimedia,Hypermedia & Telecomuunications*, June 2006, pp. 2823-2828.

Knight, A., Almeroth, K. and Bimber, B., "Observations and recommendations for using technology to extend interaction", In *Proceedings of the 18th World Conference on Educational Multimedia,Hypermedia & Telecomuunications*, June 2006, pp. 299-306.

Mazza R., and Dimitrova, V. ,"Visualising student tracking data to support instructors in web-based distance education", in *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. New York, NY, USA: ACM Press, May 2004, pp. 154-161.

"Moodle", 2006. [Online]. Available: http://www.moodle.org.

Sen, A., Dacin, P. and Pattichis, C., "Current trends in web data analysis", *Communications of the ACM*, vol. 49, no. 11, 2006, pp. 85-91.