

Supporting Collaborative Teams in Engineering Education

Francis C. Li, James A. Landay, Anthony D. Joseph
University of California at Berkeley

Abstract

Problem-based learning techniques involving small collaborative teams of students are widely regarded as being effective in teaching engineering design courses. However, the heavy requirements upon the teaching staff to support and coordinate such teams typically limit these courses to small numbers of students. In this paper, we describe the design of a computer-based tool for supporting small collaborative teams with the ultimate goal of scaling these engineering design courses to larger numbers of students. The system, deployed on laptops with wireless networking, utilizes student self-evaluations and instrumented project management tools to provide feedback of individual and team performance to the students and teaching staff. We present an example of the usage and data collected from this system during a pilot test in an undergraduate computer science design course. Based on that experience, we discuss future plans to refine the system for early detection of problem teams and to instrument asynchronous and instant messaging tools to support social network analysis of team communication.

I. Introduction

Engineering design courses commonly combine traditional lecture delivery with team project assignments, where small collaborative groups of students work on a problem related to the course material. Although this is not problem-based learning as described by Barrows, many of the same benefits arise from the addition of team project work- greater interpersonal communication between students, knowledge sharing, and a degree of self-learning of new material¹. However, while lecture delivery can scale with increasing enrollment via larger classrooms and video broadcasts, the resources required to facilitate team project work often cannot.

Monitoring the progress and performance of individual teams in a course requires significant effort from the instructional staff. Often this involves identifying social loafers and mediating personal differences between students in addition to evaluating learning outcomes. Usually, the resolution of these problems and conflicts come after they have already manifested itself as degraded team performance. As enrollment increases, however, it is difficult to scale the size of the instructional staff, a scarce resource in many engineering departments.

We believe that computer-based tools can be useful in assisting the teaching staff to monitor the progress and performance of teams in large engineering design courses. In the next section, we describe metrics for assessing team progress and performance. Then, we describe our computer-

based system for collecting and presenting those metrics to both students and instructors for awareness. We have performed a pilot evaluation that we detail in the next section, followed by a discussion of the results. Finally, we discuss future plans for the system and conclude.

II. Progress and Performance Metrics

We began our design by choosing a set of progress and performance metrics. They are:

1. Individual time spent
2. Team time spent in meetings
3. Individual action items completed
4. Team milestones achieved
5. Peer evaluation of team members
6. Private comments to the instructors

Metrics 1 through 4 have been successfully used and described by Collofello and Hart for the monitoring of teams in software engineering courses². The amount of time spent by students and teams is self-reported, and provides a rough estimate of effort and performance. Managing individual action items and team milestones provide awareness of progress as well as encourage what is considered to be “best practices” in software engineering.

Informal interviews with students working in teams confirm that they are often insecure about face-to-face confrontations with team members or the instructional staff regarding problems. The peer evaluation of team members provides a very lightweight form of feedback regarding individual contributions to the group. Each member rates their own contribution out of 100 percent, and then distributes exactly 100 percentage points to their other members. The ratings from everyone in the team (excluding the self-rating) are combined in a calculation to provide an aggregate peer rating for each member that can be compared to the self-rating.

Adding the ability to send private comments to the instructors provides an additional lightweight channel for students to present any concerns they may have regarding their team. Although email also provides a lightweight form of communication, integration with the project management tool provides a more immediate outlet when used in context with the other self-reported metrics.

II. System Overview

Our tool is implemented as a web-based client/server application. Figure 1 shows the student homepage, which is retrieved after logging into the system. The Main Menu section provides links to the individual and team report forms that collect the student-reported data. The Action Items section displays the currently active action items for the logged-in student and the other team members. The student can complete action items or add new action items from this page. Team Milestones are displayed in the upper right corner along with the due date and/or notifications that the milestone deliverables are late or done. Once each team member has

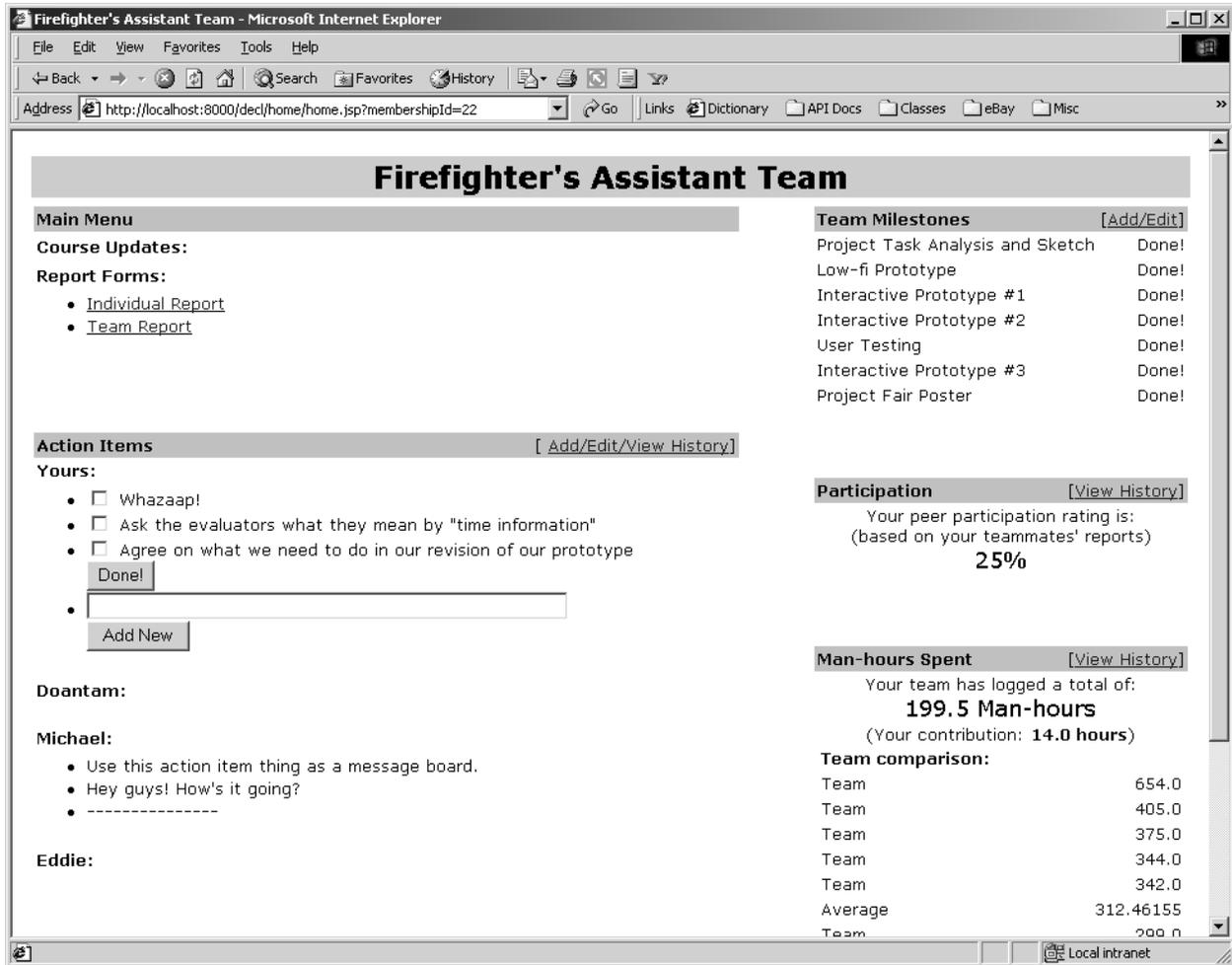


Figure 1. Student homepage showing metric summary feedback

submitted his or her individual report, the peer participation rating is calculated and displayed. Finally, all the individual and team hours are summed and displayed, with a comparison to the average and the other teams in the course. Team names are removed from the list for anonymity. Each section has an Add/Edit or View History link that will display a more detailed summary of the metric data and provide additional editing control in a new page.

Figure 2 shows the instructor course monitor homepage. The instructor homepage summarizes the metric data for all the teams in the course for a particular week. By choosing a particular team, the instructor can see summaries of the metric data for each member, as shown in Figure 3.

III. Pilot Evaluation

In the Fall 2000 semester, students in the CS160 undergraduate user interface design course at U.C. Berkeley were asked to use the system³. The course was composed of 48 undergraduate students in 12 teams of four students each, and a team of three graduate students. The

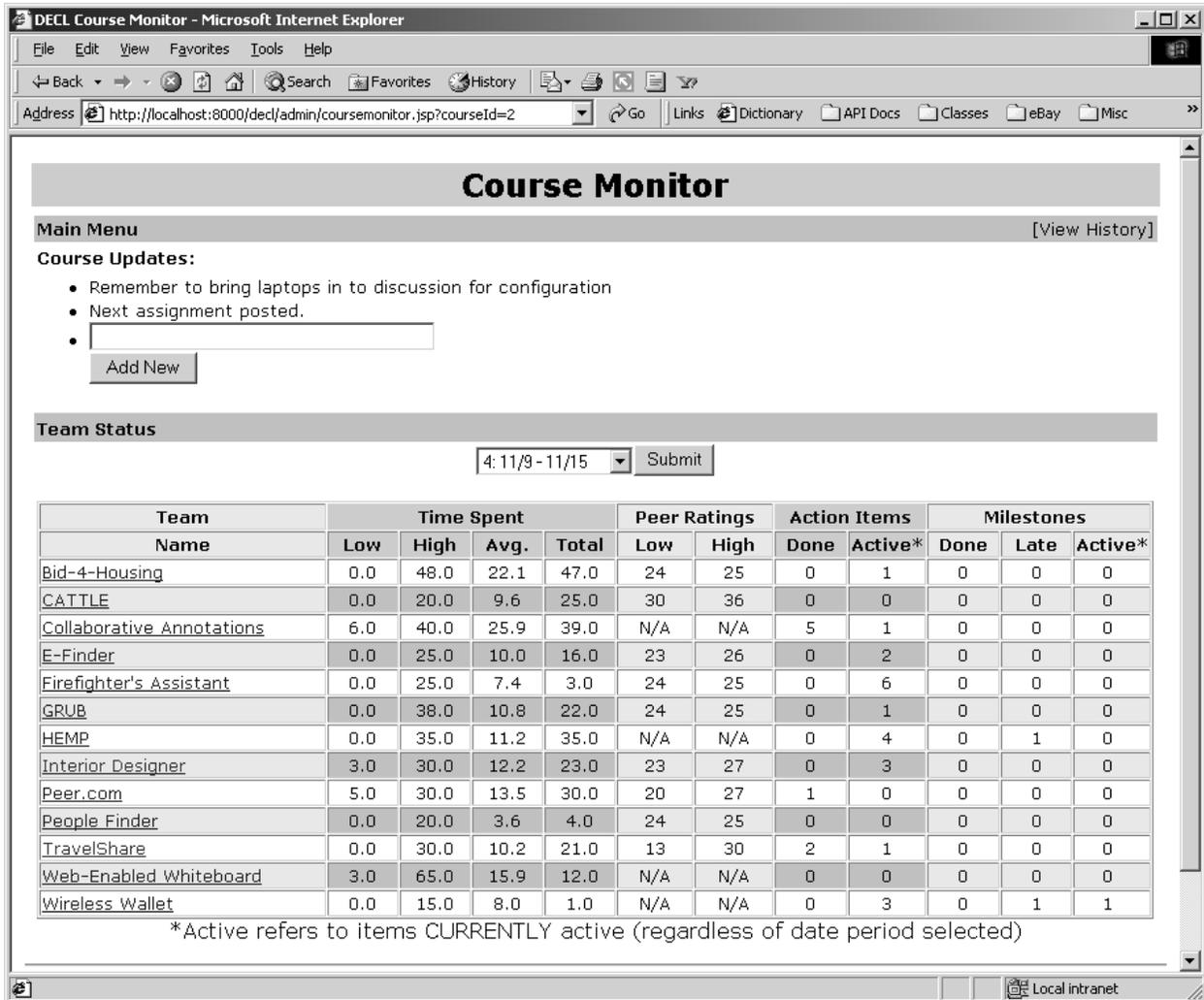


Figure 2. Instructor course monitor homepage

instructional staff included the faculty instructor and one graduate student teaching assistant. The students were asked to each submit an individual report and collectively submit a team report on a weekly basis, although the content of the reports would not affect their grades in any way. In this course, each team works on a semester-long project involving the iterative design of a user interface. The performance of the team accounts for 40 percent of each student's individual grade. The demanding workload consisting of interviewing, designing, programming, and user testing requires that the team members collaborate effectively to produce a quality end product.

In addition to the pilot evaluation of the tool, each student in the course was loaned a lightweight laptop computer running the Windows 2000 Professional operating system. The laptops were also equipped with wireless networking hardware, with coverage available throughout the computer science and engineering buildings. The laptops were deployed primarily as an experiment to alleviate the space limitations of the departmental computer labs. However, we were also interested in the potential effects additional mobility and accessibility may have on

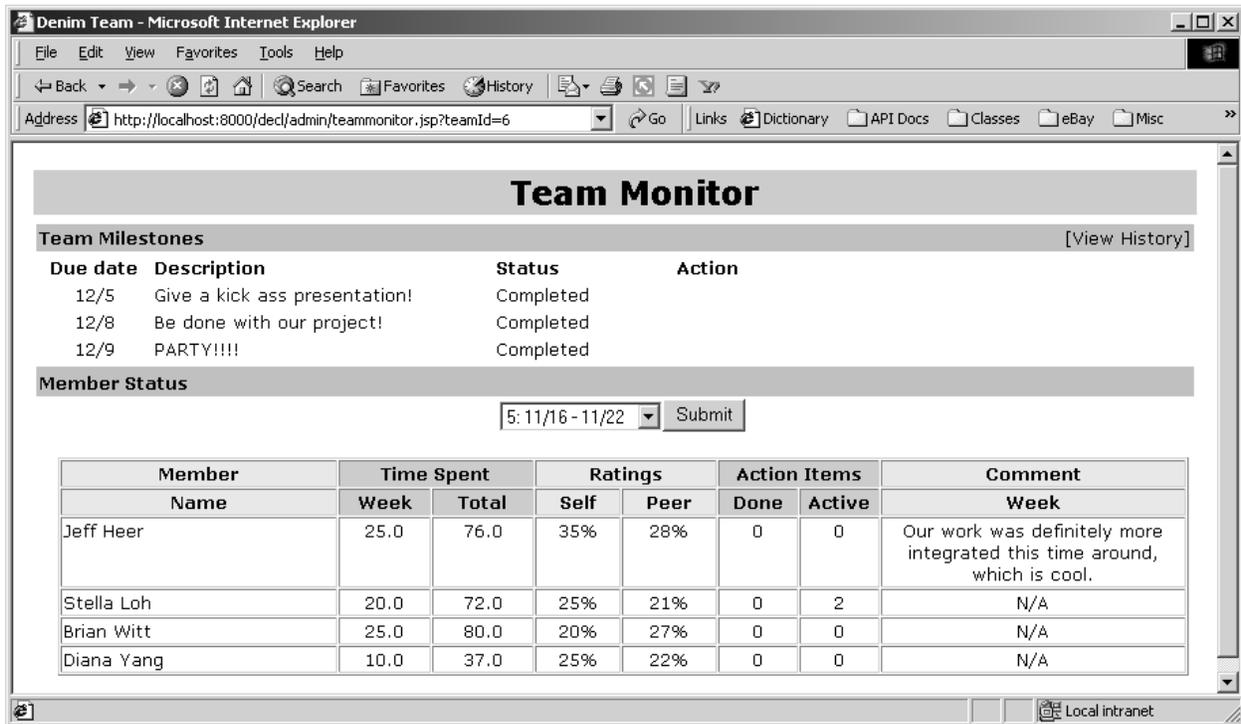


Figure 3. Instructor team monitor homepage

student work practices.

The ActiveDesktop feature of the Windows operating system allowed us to configure each laptop so that the student homepage was always available as the background wallpaper of the user interface. When connected to a network, either wired or wireless, the ActiveDesktop provides an up-to-date view of the performance metrics that are refreshed every 10 minutes. When disconnected, the most recently cached copy of the homepage was displayed.

Our goals for the pilot evaluation were to identify any design issues with the system as well as analyze the data reported and collected. In particular, we were interested in how well the self-reported metrics reflect the actual work practices of the students. Also, the student homepage provides an immediate summary of the reported data, including comparisons to other teams, which may lead to “gaming” the system so that the summaries favor particular students or teams. Finally, we conducted a number of informal surveys during and after the course to get feedback on the usage of the laptops and the system.

IV. Results and Discussion

The pilot evaluation successfully revealed a number of bugs and design issues with the system, which unfortunately affected its usage by the students. Upon first receiving and logging into the laptops, there was an immediate outcry over the use of the ActiveDesktop feature to present the

student homepage. Permissions were set such that it could not be disabled, but there was no control over the size and placement of the homepage on the desktop. As an act of defiance over that permissions policy, many students immediately resized the homepage window into a tiny box and moved it off-screen where it could be conveniently ignored. Regardless of the potential benefit of having the homepage immediately available, the desktop, considered “personal” space, is too intrusive a location.

A more significant design issue emerged from the requirement of weekly reports on a fixed schedule. Having laptops connected to a wireless network provided a greater degree of accessibility such that the students wanted to use the system more like a log or diary, submitting cumulative individual reports rather than waiting for the weekly report. The added burden of remembering the amount of time spent during the course of a week made students less inclined to utilize the reporting feature of the system. In addition, it was often easy for busy students to forget to file their report. An automated email reminder feature was designed but not implemented in time for the pilot evaluation.

Regardless of the difficulties, a total of 311 individual reports and 83 team reports were submitted over the 9 weeks of the course when the system was available. In the reports, the students reported spending an average of 10.4 hours a week individually (stdev=10.5). Figure 4 plots the average individual time spent reported for each week. The peaks at 10/19, 11/16, and

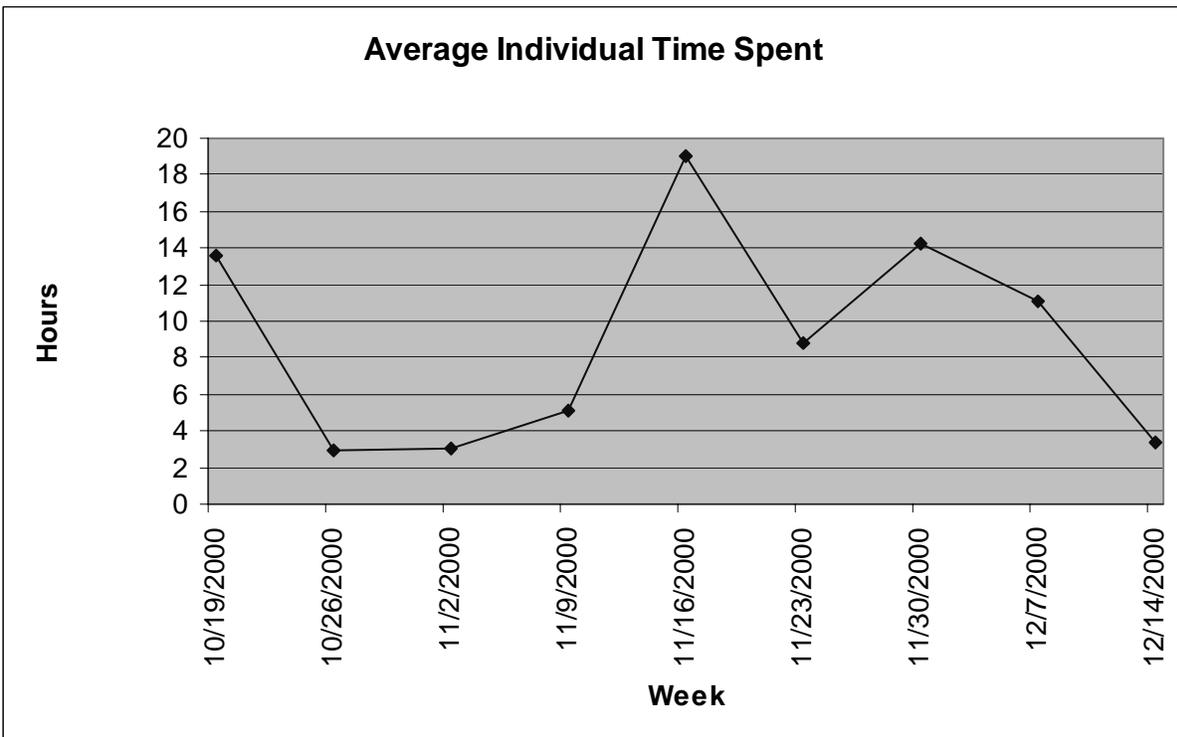


Figure 4. Average individual time spent reported per week

11/30 (weekly report periods started on Thursday) correspond exactly to weeks when a major deliverable was due in the course. This corresponds to the observed practice of last minute work and lends credence to the accuracy of the self-reported hours, at least in relative terms between each week.

As a team, the students reported spending an average of 10.2 hours together each week (stdev=12.4). The fact that this value is almost equal to the time spent individually suggests confusion in the definition of “team” time. Team time was defined as time spent in meetings and other coordination tasks on the project. However, the increased mobility provided by the laptops has changed the concept of team meetings for the students. Whereas it was previously difficult to work collocated in crowded labs, with laptops team members could meet and work side-by-side with much greater ease and frequency. In an exit survey conducted on the laptop usage, one student reported:

Working on the laptop made it very convenient to meet as a group, yet not everyone had to work on the project. Half could be working on their own thing, but when they were needed for advice or whatnot, they were always there.

Students often reported that they considered such time as “team” time and had difficulties determining a value to report. Another team described their work habits as “shifts”, where during

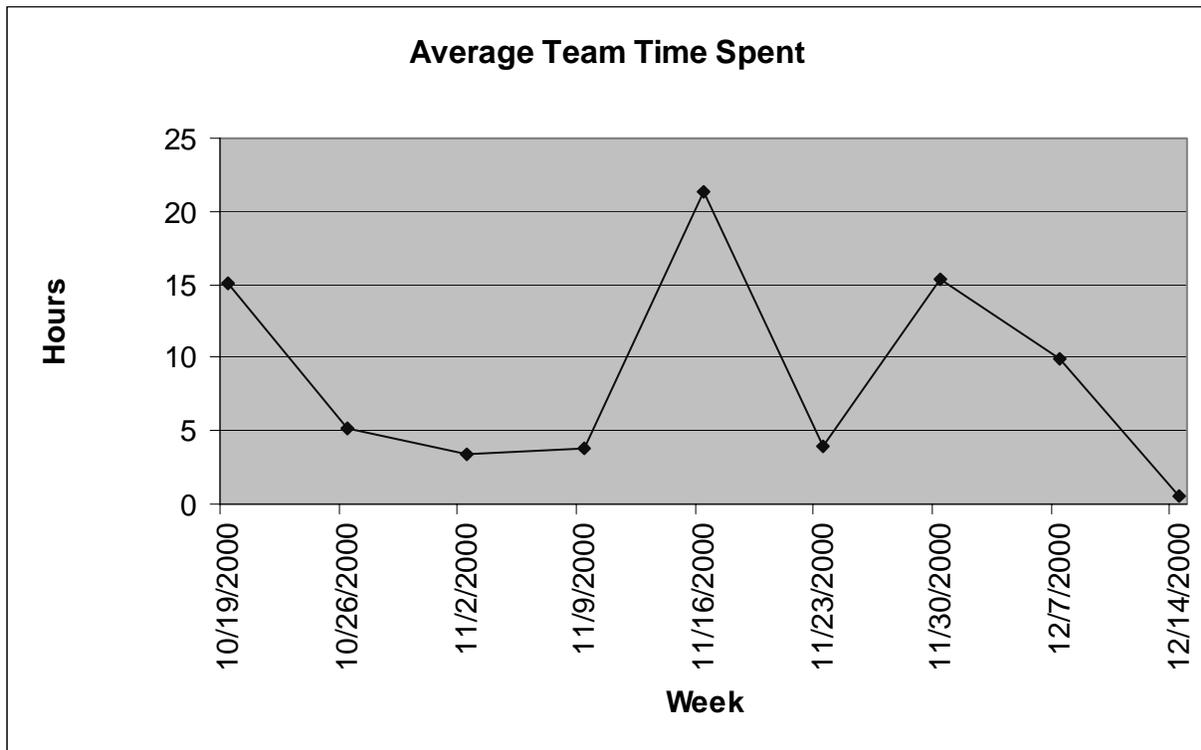


Figure 5. Average team time spent reported per week

intense periods there were always at least two members working together collocated. Again, it was difficult for the students in the team to separate the notion of individual and team time spent. Although we encourage formal team meetings between members as way of promoting industry work practices, conflicting schedules and commitments to work and classes often make such meetings difficult. The students compensate by creating their own work practices, which should be accounted for in the design of our system.

Figure 5 shows a plot of the average team time spent reported for each week. Again, the peaks correspond exactly to weeks when a major deliverable was due in the course. The average team time reported is about the same or slightly greater than the average individual time reported.

The peer participation metric requires that each team member submit their ratings in order to perform the calculation. Since many reports were forgotten, most students were not able to have a peer participation rating displayed to them on the homepage so it is unclear whether or not such information is useful or desirable. However, looking at the raw data and the comments submitted with the ratings, many students had no inhibitions about rating their team members' contributions (or lack thereof) higher or lower than average. This usefulness of this metric is still open for investigation.

When asked to rate their own contribution, students on average rated themselves as contributing

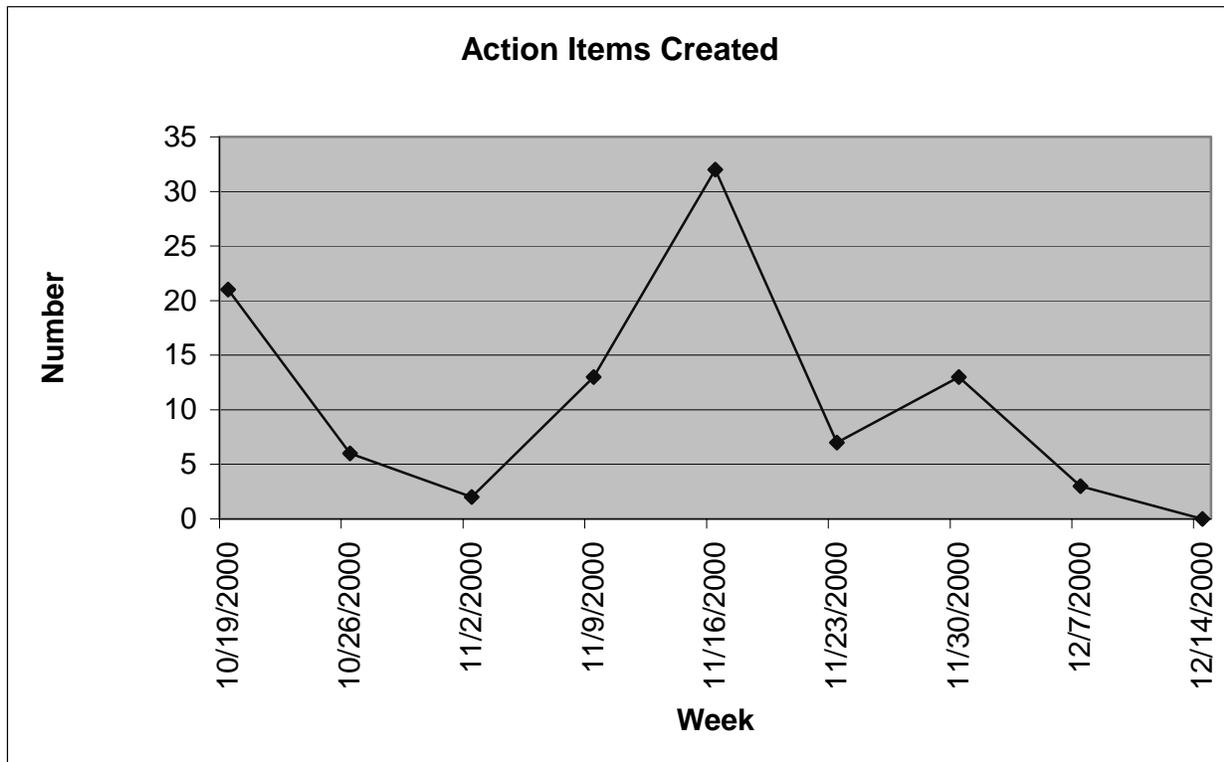


Figure 6. Number of action items created per week

27 percent of the work in the team (stdev=14.6). Given four students per team for all groups except one, this is very close to the equal contribution of 25 percent. The high standard deviation is largely due to confusion over contribution ratings during weeks when there were no team deliverables assigned. Most students simply rated their contribution as zero, while some submitted 100. Looking at the raw data and the comments submitted with the self-ratings, many students were very open in describing their contribution, often citing other course and work commitments when their contribution was low.

Free-form comments at the end of each report form were intended to give students an opportunity to reflect upon their work at the end of each week. In the first week in which the system was deployed, a number of comments were submitted suggesting potential problems:

Since it's so difficult to meet with everyone in the group at once, organization is a bit lacking.

Our group (and I am no exception) has a problem working together. We gravitate towards portioning out work and doing our assigned parts, emailing parts around once complete. It's not as much a team environment as a group environment. This is primarily because we have such different schedules and requirements, different working styles, and lack of common inspiration for our mission statement.

I was going to come talk to you all about this, but I haven't gotten around to it - I've found that it's hard for me to get my group to actually do work on their own - I have to be there to keep things going, or do the work myself, otherwise things don't seem to get done right... I don't know if there's much you can do about it, and I'm hoping that for the next few projects I can get everyone to do a fair share.

The first two comments were submitted by members of the same team. The last comment suggests the utility of having an immediate outlet for free-form reflection as part of the reports. In each of these cases, the students were individually contacted and the problem was discussed. It cannot be determined if this early intervention was productive, but the corresponding teams did not report any problems in following weeks and produced quality work.

Looking at the project management tools, only 23 students used the action items feature, creating 128 action items, of which 98 were completed, 8 were deleted, and 22 were not completed. Figure 6 plots the number of action items created during each week. Again, the peaks correspond to weeks where major deliverables were due. In observing the content of the action item descriptions, however, a few students used the feature to send short messages to their team members, since each member's action items are prominently displayed on the student homepage. Team message boards were considered in the design of the system, but not included due to time constraints. We consider the addition of message boards in the next section as we discuss future work in observing communication patterns.

Only seven of the 13 teams in the course tried using the milestones feature, defining only 28 milestones in total. The system starts out with no milestones defined, and only one team entered the major deliverable due dates into their schedule. Since the deliverable due dates are very clearly presented in the course through handouts and the course homepage, there was likely little benefit for the students to add them. More importantly, none of the milestones entered represented clear incremental steps between major deliverables. This may be an effect of having major deliverables almost every two weeks in the course, but better instruction on its use and possibly seeding the milestone schedule in advance may encourage the use of this metric.

V. Future Work

The pilot evaluation revealed a number of design issues with the system design, some related to the introduction of laptops with wireless networking into student work practices. We plan to iterate upon the design of the system to accommodate many of the issues discussed as well as improve its robustness and reliability. In addition, we have interests in monitoring communication patterns between team members.

Many difficulties in team collaboration are often due to poor communication between members, as seen in some of the comments quoted previously. An area for investigation that we are interested in pursuing is the observation of both synchronous and asynchronous messaging between students for social network analysis. Such an analysis would provide a metric of communication “connectivity” between students in teams, and may be useful in identifying social loafers or other dysfunctional members. For example, a few students used the action items feature on the homepage for posting short messages to each other. By providing message board services as part of the homepage, we can facilitate asynchronous communication and monitor the participation of each member of the team via their postings.

An additional source of communication information is synchronous “instant” messaging. Each student in the course was asked to sign-up for a Yahoo! Instant Messenger account and each laptop was configured with the corresponding messaging application. The main incentive for students to sign-up for the account was to have an additional means of contacting the instructional staff. Significant usage of instant messaging was observed between students and the teaching assistant, primarily for asking questions on course material and assignments. In addition, a fair amount of usage was observed between students in the course. We are interested in deploying an instrumented instant messaging application, based on the Ninja IM service developed at U.C. Berkeley, for logging and analysis of these communication practices.

Finally, we are actively recruiting faculty and student researchers in the U.C. Berkeley School of Education to participate in this project. In particular, we are seeking their assistance in planning a more thorough assessment of learning outcomes using the system. As the system becomes more robust, we would like to deploy it in incrementally larger courses in the computer science department.

VI. Conclusion

We have described a computer-based system for monitoring the performance and progress of student teams in engineering education. The system utilizes student self-report and instrumented project management tools to collect data supporting a set of metrics of team performance and progress. Our pilot evaluation of this system, as used in an undergraduate design course, has shown that the data reported and collected using the system does reflect the work habits informally observed by the instructional staff. In addition, a number of design issues have been discovered that reflect changing work practices introduced with the addition of wirelessly networked laptops into the course.

VII. Acknowledgements

This work was funded by grants from Intel Corporation and the National Science Foundation.

Bibliography

1. Springer, L., Stanne, M. E., & Donovan, S. S. Effects of Small-Group Learning on Undergraduates in Science, Mathematics, Engineering, and Technology: A Meta-Analysis. *Review of Educational Research*, Spring 1999, 69(1). 21-51.
2. Collofello, J.S. & Hart, M. Monitoring Team Progress in a Software Engineering Project Class. In *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, 1999. 7-10.
3. Fall 2000 CS160 course homepage, <http://bmrc.berkeley.edu/courseware/cs160/fall00/>

FRANCIS C. LI

Francis Li is currently a graduate student at the University of California at Berkeley in the Department of Electrical Engineering and Computer Science. His primary interests are in human-computer interaction, user interface design, computer mediated communication, and collaboration tools. Francis received a B.A. degree in Computer Science from the University of California at Berkeley in 1998.

JAMES A. LANDAY

James Landay is an Assistant Professor of Computer Science at the University of California at Berkeley. His primary research interests are in human-computer interaction, user interface design tools, pen-based user interfaces, end-user programming, mobile computing, and other novel uses of computer technology. Dr. Landay received a B.S. degree in Computer Science from the University of California at Berkeley in 1990 and a Ph.D. from the School of Computer Science at Carnegie Mellon University in 1996.

ANTHONY D. JOSEPH

Anthony Joseph is an Assistant Professor of Computer Science at the University of California at Berkeley. His primary research interests are in mobile and distributed computing, wireless communications (networking and telephony), and smart spaces.