INCREASING STUDENT PARTICIPATION IN A NETWORKED CLASSROOM

Thomas Gross, Levente Szekrenyes, and Cristian Tuduce[1]

Abstract—*Many students own notebooks, and many lecture halls are equipped with a wireless local area network, so student notebooks provide a convenient platform for various in-class activities. We report here on a simple tool that uses this platform to increase student participation. An instructor can perform instant in-class assessments, solicit questions about the material, or gauge the student's perception of the lecture's speed. Responses are collected without revealing the student's identity and stored on the instructor's machine or a remote server. If appropriate, the responses are electronically tabulated (e.g., multiple-choice questions or self-confidence surveys). The instructor always maintains control and decides when an assessment or query is posted, how long the students are given to respond, and if summaries are shared with the class. Also, even in the most computer-savy environment, not all students bring a computer to class, and we discuss how to deal with this issue.*

Index Terms – In-class assessment, polling, classes with large enrollment, wireless local area networks.

INTRODUCTION

Polling students in a lecture is a simple technique to get the students to actively participate in a class. The instructor asks a question and asks students to "vote" on the question. Polling is an informal public assessment that provides instant feedback to students and instructor alike. In large classes, it may provide one of the few opportunities during a lecture that allows each student to voice his or her opinion on a subject. Polling is usually done by counting votes, e.g., a show of hands, or measuring the applause level[4]. However the public nature does not appeal to all students, and reading the minds of the "silent majority" is difficult, as many instructors experience when facing a large class.

Over the last couple of years, the use of notebooks by students has increased significantly. Many institutions encourage and/or subsidize student ownership of notebooks[2]. Originally, portables were primarily used to take notes. But as wireless local area networks (WLAN) become common, these notebooks allow students to connect during class to a server (and/or the Internet, depending on the the institution's WLAN setup or policies)[3]. At this time, there are three major flavors of notebooks: PCs with some kind of Microsoft operating system, PCs with Linux, and Macintoshes (overwhelmingly used with Apple's OS). All systems have in common that they support a range of browsers. The general-purpose nature of these notebooks and their connectivity features make them interesting platforms for all kinds of in-class activity that require a student's active participation.

A classroom assessment is a technique to obtain feedback about student learning that is more formalized than polling. Assessments require the active participation of a class, and there exists a list of well-known (and often used) techniques to perform assessments in a class. Angelo and Cross have collected a large number of classroom assessment examples and scenarios[1]. Activities to solicit such feedback differ in form and intent from graded tests or exams, which serve to determine a student's performance.

Frequent feedback (to instructor and student alike) during a semester is desirable. A student finds out where he or she is standing, an the instructor may make adjustments in course content or delivery. Angelo and Cross[1] document the positive effects of classroom assessments. Students feel more involved and appreciate that their opinion counts. Instructors do not have to rely on guessing to obtain information on student learning. In small classes, with 10 – 20 students, many forms of in-class activities can be included. Once the enrollment is 100+ students, there are fewer options and assessments provide a well-tested approach to structure interactive participation.

Assessments are usually done anonymously and are therefore often voluntary (that some students may not participate, e.g., because of a lack of interest or an absence, does not diminish the value of an assessment as long as representative responses are obtained). Traditionally, assessments are done with paper and pencil and are processed by the instructor off-line, after class.

One impediment to performing assessments in a class is the cost of evaluating the assessment – especially with larger classes even a simple task like tabulating responses to a few multiple-choice questions is time-consuming and cumbersome. This is bad since providing the results of an assessment to students is an important part of many classroom assessment techniques[1].

In this paper we describe a tool to perform and evaluate assessments in class, using the students' notebooks as entry pads and the instructor's notebook (or a server) to collect the responses. The tool has been implemented and used for 2nd year students (the university had targeted this class for a pilot program to experiment with notebooks). We first discuss the situation that the lecture hall is covered by the institution's WLAN. Later we discuss how this requirement can be avoided, at the cost of an increase in complexity.

DESIGN CONSIDERATIONS

The system must be easy to use for students and faculty – everybody is already busy enough. The basic architecture therefore includes a Web server to post questions/instructions and a client applet that is downloaded from the server. The Web server can run either on the instructor's notebook or on any other machine that is reachable from the lecture room.

[1]ETH Zürich, CH 8092 Zürich, Switzerland

Client

The mode of operation is then that the student downloads the assessment applet from the Web server at the start of class. The applet polls the server for new assessment questions/instructions. Each assessment opens a new window with two panes: one that displays the instructions and one that takes the student's response. In the current implementation, the interaction is limited to simple multiple choice questions. After a student has made a decisions, s/he selects the appropriate choice and confirms it by clicking the "OK" button.

Master

A central issue is that the instructor always maintains control of the assessment process. Assessment instructions are stored in a file that is usually prepared ahead of class. The instructor decides when instructions are made available, for how long responses are collected, and once an assessment is closed, the server ignores further responses. Also, there exists the option to send to all those that responded the correct answer [if the kind of assessment provides for one].

Any new tool that is introduced into a class carries the risk to complicate the instructor's tasks. Not only must the instructor operate the tool, he or she must be prepared to deal with errors or problem cases. To minimize the chance that the instructor's input causes problems by accident, all critical steps require an extra level of confirmation. In the interest of avoiding information overload, after a question has been posed, the instructor's interface shows for this question only the number of responses and the breakdown into options.

In addition, the master maintains extensive logs (when did the responses come in, which hosts connected how long, how did each host respond) so that more extensive analysis is possible off-line (e.g., to check if there is a correlation between answers to different questions). Such analysis could be done in class in real-time, but most instructors may not want to spend the time on such activities in class. However, such analysis is not part of the tool proper and must be done outside (using a wide set of specialized tools that are available).

Discussion

This "master–client" design has several (practical) advantages:

- Minimal setup before class: the instructor must set up a Web server and connect to the server if it isn't on her/his notebook.
- No support or involvement of a facilities or infrastructure group is needed. (Many OS provide detailed instructions on how to run a Web server.)
- Students engage in a fairly well-known activity (download a Java applet). The student can use his or her preferred browser, as long as it supports a current JVM.
- The student's identity is hidden if the mapping from user IDs to IP addresses is not revealed. (This mapping is dynamic, and the logs are usually visible only to network administrators.)

```
<QUESTION>
Race conditions
<ANSWERS>
Are found only in message passing programs.
* Can arise when two processors write
      the same variable.
Are used to determine horse race handicaps.
Cannot arise in explicitly parallel loops.
```

Fig. 1. Instructor's input. * indicates the correct answer.

- Platform independence is maintained both for instructor and student, a single master can serve all kinds of student platforms. The instructor is not tied to a specific kind of host OS.
- Since the applet is downloaded each session, updates are easy (and do not require extra participation by the students). This aspect won't matter that much now that the system is stable but was appreciated during the development and validation phases.

IMPLEMENTATION

Figure 1 shows the sample input that must be prepared by the instructor for a simple question (the instructor explained that students have to choose the most appropriate answer).

Figure 2 shows the student's view for the question from Figure 1.

Figure 3 shows the instructor's view. The instructor's window consists of several panes: on the left, a question and its list of possible responses is seen. Additional questions read from a file are available from a pull-down menu. In the center, we see the list of questions already sent; on the right, IP addresses of all hosts that have established contact with the server are shown (since this snapshot was taken in the lab, there is only a single client).

The implementation gives each student a single opportunity to submit a response, there is no way to "recall" or correct a submission. Furthermore, we attempt to prohibit double responses from the same student (notebook). "Double voting" would be possible if the student were able to load the applet twice, e.g., from different browsers or windows.

EXPERIENCE AND ISSUES

Limitations

In its first incarnation, the tool tabulates only multiple–choice questions. The closed-end format of this kind of question allows only a limited form of interaction. Other kinds of questions (ordinal input, Likert-scale ratings, categories) fit also the interaction format that is embodied in this tool. The current tool's mode of interaction is "one question at a time", so a student must answer to a list of questions in the order that the instructor sends the questions to the clients.

The form of responses that is handled can also be extended to programs. One colleague has asked students to write a short assembly program to solve a specific problem and the

Fig. 2. Student's view for a single question.

Fig. 3. Instructor's control of the tool.

TAs analyzed the solutions in real-time during class. However, there exist a number of tools to provide some form of automatic analysis of programs. The (partial) correctness can be evaluated using a test suite, and the efficiency can be measured by timing the execution. (If the instructor plans to execute a student-supplied program on his/her machine, it is a good idea to check submitted programs for inappropriate system calls!) Since in-class assessments are short, any program written by students must be short. Obviously this type of processing will not work for complex problems and those situations that call for more detailed feedback and analysis.

Student-driven interaction

A multiple-choice query is posed by the instructor and answered by students. If the students see the responses of the class while the question is still in their memory, the loop is closed and the interaction, although not personal, provides the student some feedback. To allow the students to initiate interaction with the instructor, an applet can solicit questions from the students which are then sent to the instructor's computer. It is up to the instructor to decide which question should be answered in class (maybe the instructor reads the list while students are given a problem to solve).

The same setup can be used to allow the student to submit text, e.g., in a "1-minute summary". Of course, for such a question it would no longer be possible to provide instant feedback since an automatic analysis may be problematic. However, there are still benefits obtained from collecting the responses in digital form (they can be forwarded, kept in an archive, or an instructor can implement his or her own off-line filter).

Practical considerations

Not all students own a notebook. It is important to realize that an assessment taken with this tool produces a sample, not a definitive answer. And there is of course the fundamental concern that bias is introduced by using notebooks to solicit responses. Detailed follow-up studies will have to address this issue.

To also involve those students that don't have a notebook, we project the instructions (using an overhead projector) and ask students to think through the problem. At the end of the work period, students without computers can be asked for their solution/opinion.

Preparation of instructions beforehand is advisable. However, the simple format minimizes the ahead-of-class effort. Once we had a distinguished external guest speaker in class and we asked him to provide us with a few questions the students could be asked at the end (to see what they remembered from the guest lecture). The guest complied and during the lecture the questions were keyed in, and an assessment could be made at the end.

To allow the dynamic inclusion of issues that come up during class (when most instructors will not want to type in even the simplest question), we prepare a set of "generic templates". Those templates expect that the instructor provides the instructions on an overhead projector (or blackboard). The generic question list 2 (yes/no), 3 (A/B/C), or more options, and the student is asked to select the most appropriate response. The instructor then writes the concrete question and responses on the overhead.

The fact that not all students bring a notebook to class is not necessarily an impediment to using this tool. Many assessment techniques ask a (small) group of students to discuss a problem and then to come up with a common answer [1]. As long as one student in the group has access to a notebook, there is no problem.

Rewards

Many instructors report that student participation in any activity increases when students receive some reward for their effort. Since assessments are (usually) anonymous, participation is rewarded, not providing the "right" answer. When students turn in their assessment, their name is recorded or they receive a piece of paper that confirms their participation. It would be easy to add to the current tool the ability to send each client a certificate that confirms the owner's participation (maybe issued only if all or at least a pre-set number of assessment steps are done). The certificate could be printed and turned in to the TA for some form of "credit". Although the certificates could be copied, if each has a unique encrypted key, it would be easy to identify copied or forged certificates, at least with assurance that is appropriate for this scenario.

However, as long as the use of a notebook is not mandatory and not everyone can participate, such a reward system appears questionable and we rely on appealing to the students to participate out of enlightened self-interest. If students learn about the results of the assessments, this provides usually motivation enough. Especially in large classes, students are very eager to find out how that they are not alone with their misunderstanding of a topic and appreciate any effort by the instructor to give them a voice.

Student feedback can of course also be integrated into a course management or e-learning platform for a course (e.g., WebCT and other tools provide various mechanisms to gather feedback). However, as Angelo and Cross argued [1], devoting a few minutes at the end of a class to perform the assessment allows an instructor to obtain immediate feedback of those who attend(ed) the lecture. In-class assessments should not be seen as an alternative to other forms of assessment; in-class assessments add another option to an instructor's tool chest, and the instructor must decide, based on the teaching goals, when and where to use a technique. Furthermore, most course management systems (e.g., WebCT) require some form of authentication, making it harder to gather assessment data anonymously.

Stream queries

When using the assessment tool to collect responses to multiple–choice questions, we replace filling out a paper form with clicking on buttons. However, the use of on-air notebooks also allows interaction styles that cannot be realized with paper-based assessments. E.g., we can set up a *stream query* that allows a student to continuously rate the difficulty of the material (or to indicate when he or she feels the speed is too slow or to fast). In a stream query, multiple responses are possible, and the instructor's window shows the responses collected during the last n minutes. Future studies will have to investigate if such fine-grained techniques help or hinder student learning. But this tool provides a technology base that makes such studies possible and should be seen as enabling technology, not as a definite answer.

Network realities

There was some concern on the local 802.11b WLAN's ability to support a large class and to handle the traffic generated by the assessment (download of the applet, download of the questions, upload of responses). However, in our environment (approx. 50 notebooks/class in a class of 140), we are far from reaching the network limits. We extrapolate that the system will work for mid-100s of notebooks but (un)fortunately have no practical experience with classes of this size.

Not all lecture halls are equipped with a WLAN (or some institutions may not even have a WLAN). In this case clients and servers have to manage their communication differently, using an "ad hoc" network. (An ad hoc network operates without a central infrastructure, the notebooks are node and router (forwarder) in the network at the same time.) Ad hoc networks are a topic of active research in the networking community, but there exist already a number of software packages that can be used at this time.

Support of ad hoc networking differs for the various operating platforms. We have used an ad hoc network for involving the Linux notebooks (approx. 80%). At this time, clients and server must install a kernel extension, and some instructors or students may be reluctant to do this. However, as ad hoc networking is better understood and supported, more user-friendly solutions will be provided for all common platforms.

CONCLUSIONS

Instructors that want to perform a rapid assessment can benefit from this tool. Also, instructors that use some form of polling will find the tool helpful. By automating the processing of the responses, the tool described here removes one of the hurdles cited by other practitioners to classroom assessments. Of course, counting the responses is only a first step in analyzing the result. Every instructor will have to decide how and when to share the results with a class.

We do not advocate instant assessments as the only technique to obtain feedback. However, where appropriate, the automatic collection of responses provided by this tool can save a lot of instructor (and/or TA) effort.

Not all types of classroom assessments are equally well-suited to a simple tool as described here. But the tool is extensible, and instructors that want to automate other kinds of assessment are welcome to continue.

The process is not perfect – students are free to participate and so the response is only as good as the instructor obtains coverage. But most students appreciate an instructor's effort to gather information about their learning experience. And as computer scientists struggle to involve the students in courses, the tool presented here provides a simple, zero-cost enrichment of the student's learning experience.

The tool can be downloaded from `www.lst.inf.ethz.ch`; the conditions of the GNU Public License apply.

ACKNOWLEDGMENT

We appreciate the assistance of A. Brunner of the ETH computer services division.

This work was funded, in part, by a gift from Intel Corp. Additional funding was provided by the NCCR "Mobile Information and Communication Systems", a research program of the Swiss National Science Foundation, and by the Provost's fund for innovation in teaching (FILEP).

REFERENCES

[1] Angelo, T. A., and Cross, K. P. *Classroom Assessment Techniques, 2nd Ed*. Jossey-Bass Publishers, 1993.

[2] Carlson, S. Universities find wireless systems bring them convenience and savings. *Chronicle of Higher Education* (Oct 2000).

[3] Grace-Martin, M., and Gay, G. Web browsing, mobile computing and academic performance. *Educational Technology and Society 4*, 3 (2001), 95–107.

[4] Wolfman, S. A. Making lemonade: Exploring the bright side of large lecture classes. In *Proc. 33rd SIGCSE Symp.* (2002), pp. 257–261.