AC 2008-1611: ACTIVE LEARNING ACROSS THE COMPUTER SCIENCE CURRICULUM

William Birmingham, Grove City College

Vincent DiStasi, Grove City College

Active Learning Across the CS Curriculum

1. Introduction

The general idea of active learning is to engage students during lecture with a variety of things that *actively* involve them in the material being presented. Through active engagement, students are highly encouraged to focus attention on the lecture. While active learning holds great promise, it also raises several important issues: designing materials for lecture that incorporate active learning while keeping lecture preparation reasonable, managing the classroom, and ensuring that the necessary amount of material is presented during lecture (time spent on activities is time not spent in lecture). While there is still much research being conducted to address these issues, many techniques have proven successful.

With proper technology support, active learning techniques become much easier to apply. In particular, specialized software helps to both manage the classroom and to create lecture materials. The Tablet PC, with the functionality to easily and quickly *write* equations and diagrams so important to the CS curriculum, is tremendously important in supporting active learning. [1,2]

In this paper, we describe how the Tablet PC and specialized software is used across the CS curriculum, from our introductory programming classes through our most advanced courses for seniors. By using software to support active learning, faculty can use a variety of methods to engage student during a lecture, moving far beyond simple questions and answer. These Tablet PC/software systems allow students to privately ask the instructor questions, to report their status, and, under faculty control, ask questions of other students through “chatting.” More importantly, however, faculty can pose problems to students, have the student generate solutions, and return those solutions to the faculty member, who can display selected responses or evaluate them in real-time.

By using these different methods to engage students in problem-solving during lectures, we have found that not only do they participate directly in the lecture, but the faculty can make “real-time” assessment of how well the students are following. This allows the faculty member to adjust the lecture, such as increasing the pace of the lecture if the students are keeping up, to adding new material if the students are falling behind.

In this paper, we describe the hardware and software systems we use for active learning. We also describe the pedagogical methods we have developed over the past three years that applicable to a wide variety of CS and engineering classes. Finally, we present summary longitudinal data from both students and faculty showing the strengths and weaknesses of active learning.

2. Hardware and software

Grove City College has a 1:1 mobile computing program, now in its second decade, where each student receives at the start of his or her freshman year a computer. For the past four years, the computer has been a Hewlett-Packard Tablet PC. Currently, there are about 2500 Tablet PCs used by students and about 120 used by faculty.

Complementing the hardware program, the college provides software to support the entire CS and engineering curriculum from integrated development environments (IDEs) to applications such as MatLab and Maya. Moreover, the Tablet PC has a variety of software applications that are pen-aware, such as Microsoft® Word and OneNote™ which directly support note taking (and enhance it by allowing notes to be automatically indexed by desktop search engines). Most importantly, the Tablet PC allows the student to use the modality with which he or she feels most comfortable: typist, writer or a combination of both.

Finally, every seat in classrooms at Grove City College have network access through wired and/or wireless connections. In addition, all students have network folders for backup, shared project space and course folders.

While most faculty expected that Tablet PCs would be used in the classroom for note taking, running demonstration projects, and so forth, this was not occurring at the rate in which we expected. We observed over the years that Tablet PCs (and the laptops that preceded them) were primary used in the classroom to complete assignments using various software packages, as well as gaining access to electronic content through databases, etc. In the classroom, the majority of students, continued to take notes with pen and paper due mainly to the type of content that was being delivered. This used seemed to countervail the expectations we had for substituting the Tablet PC for blackboards.[3,4]

To better use the capabilities of the Tablet PC in the classroom, the college began a research program to evaluate DyKnow® Software (Vision and Monitor)[5], Classroom Presenter[6], and Microsoft® Research ConferenceXP[7]. We collectively call these classroom learning systems (CLS). We have deployed DyKnow® Software to 39 classes with 17 faculty and approximately 800 students. These courses cover a broad range of departments at the College, from engineering and science to the humanities. Within the Computer Department's course offering, 11 of 15 courses used either DyKnow or Classroom Presenter in the Fall, 2007, term; this is the typical usage over the past year.

3. The Classroom

DyKnow and Classroom Presenter allow the lecturer to deliver his or her presentation directly to the students Tablet PCs in real-time. That way, when the student leaves the lecture, he or she has the *complete* notes for the day on his or her machine. There is no need for a faculty member to upload PowerPoint to a server and a student to download.

Since the lecturer is using a pen, he or she is free to write, sketch, use symbols, write equations, and so forth as on a blackboard, yet the material is captured on the student's machine immediately. The students can write personalized notes in real-time on the lecture material directly on the delivered slide or in a private "notebook." These systems also allow the student to easily and privately provide feedback on the level of their understanding of the material while it is being presented, ask the instructor questions through a participant-to-moderator chat, and also capture course content not only from the lecturer, but through slides submitted by other classmates that the lecturer may make available. It allows for interactive and collaborative teaching and learning while also providing a means of classroom management.

In the following sections, we show how lectures are delivered and received. We also show how we use various mechanisms for fostering active participation in the lecture. Due to constraints on the paper length, we will base examples on DyKnow, but Classroom Presenter provides similar capabilities.

Upon logging into the DyKnow client, either the instructor or student specific options are displayed based on the identity of the user. During lecture, these clients are linked through a server, which is also used after lecture to store and retrieve notes. Alternatively, notes may be stored on the Tablet PC or any removable storage device. In the sections that follow, we provide screenshots from both the instructor's client and the student's client.

3.1 Lecture: Faculty Perspective

In this section, we show how CLS can be used for two of the most common lecture activities.

3.1.1 Diagrams, Equations and Text

We have found that a mix of notes and prepared graphics is an effective way of delivering lecture material. This method is particularly important for CS classes, as complex drawings are often needed to explain concepts. Figure 1 shows an example of a prepared slide with markups that were made during lecture. The lecture is from Introduction to Computer Game Design and Development and was on line clipping.

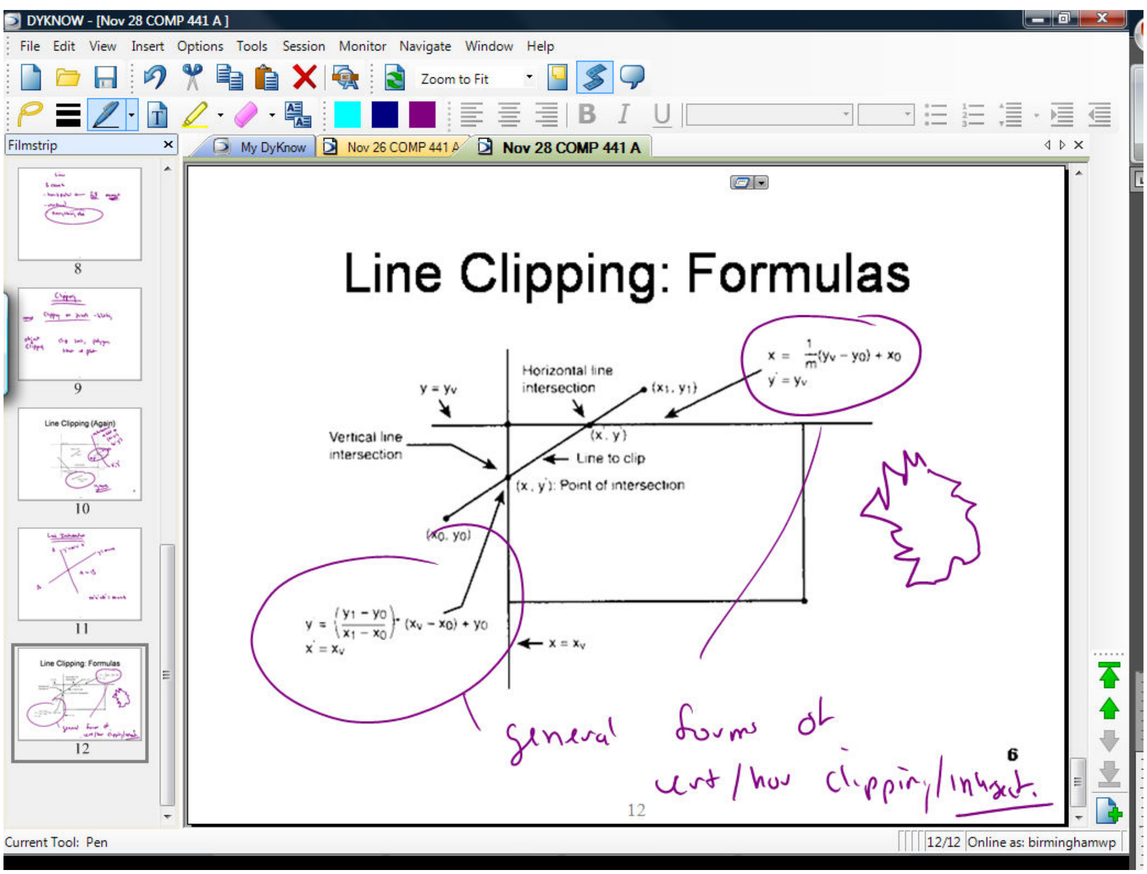

Figure 1: Mixed PowerPoint and notes.

The hand written and drawn comments were made during the lecture. The polygon with numerous points was drawn in response to a question about a figure that would cause the clipping algorithm problems (i.e., a pathological case).

Figure 2 provides an example of a “panel” that was written completely by hand during the lecture. In this case, there is a mix of drawing and equations (both are relatively simple, but nonetheless demonstrate the capability). This panel was developed over about a five minute interactive portion of the lecture, where the lecturer drew the coordinate plan and the slope/intercept line equation and asked the students to draw the resulting line as it would appear on a display screen. The point of the example was to demonstrate the problem with the obvious method for line rasterizing, and motivate the need for a true rasterizing algorithm, like Bresenham’s Algorithm.

This type of example would have been drawn on a blackboard in the past. The order in which points are added to the line in this slide is very important. A blackboard would not adequately capture this ordering, as it cannot be replayed, nor can a student’s notes be replayed. However, in DyKnow, the student can “play" the slide, where the software will replay each pen stroke in the order it was made.[1]

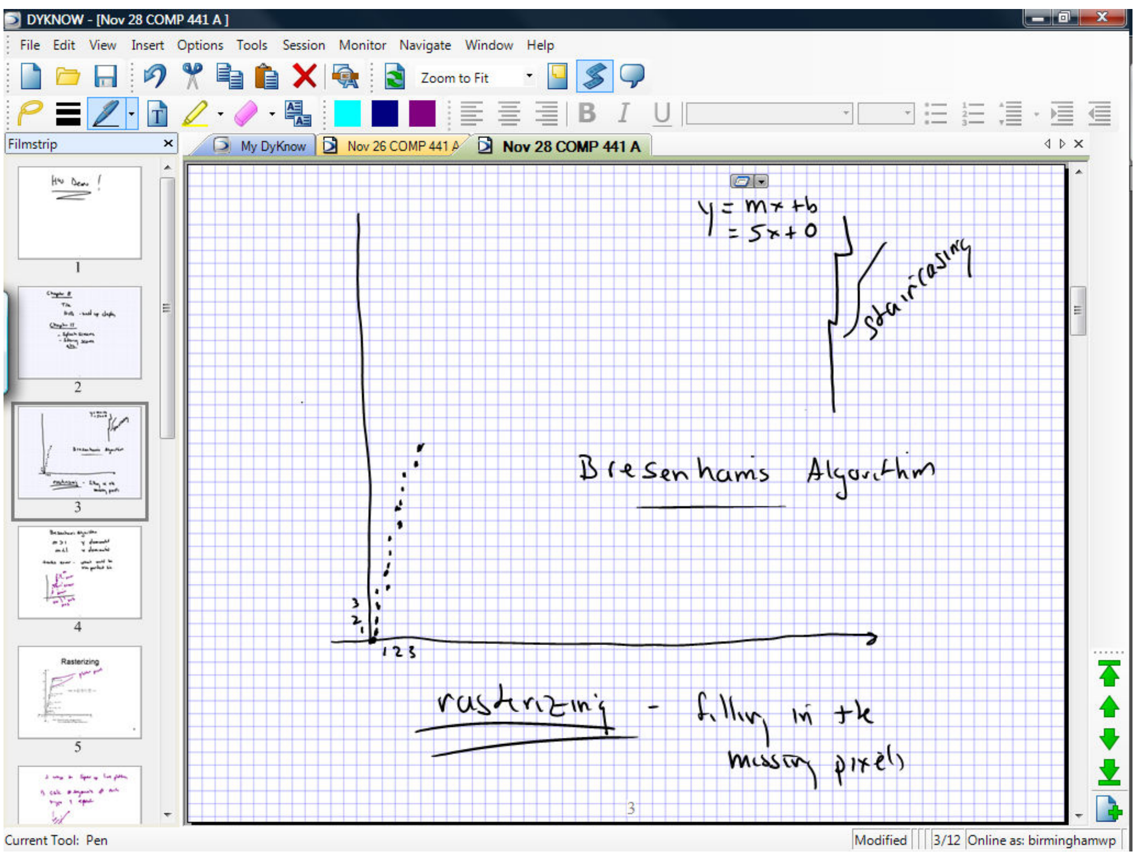

Figure 2: Hand-drawn figure and equations.

[1] We do note, sadly, that the lecturer’s mistakes are able to be replayed, too. Thus, erasing and rewriting will not work as it does on a blackboard.

Figure 3 shows a panel that is a PowerPoint slide with some minor inking made on it. Here, the lecturer was trying to show how to speed up rasterizing by precomputing "chunks" of line segments. As we show in the next section, this slide was given to the students as an assignment in class and we will show an example of what they returned.

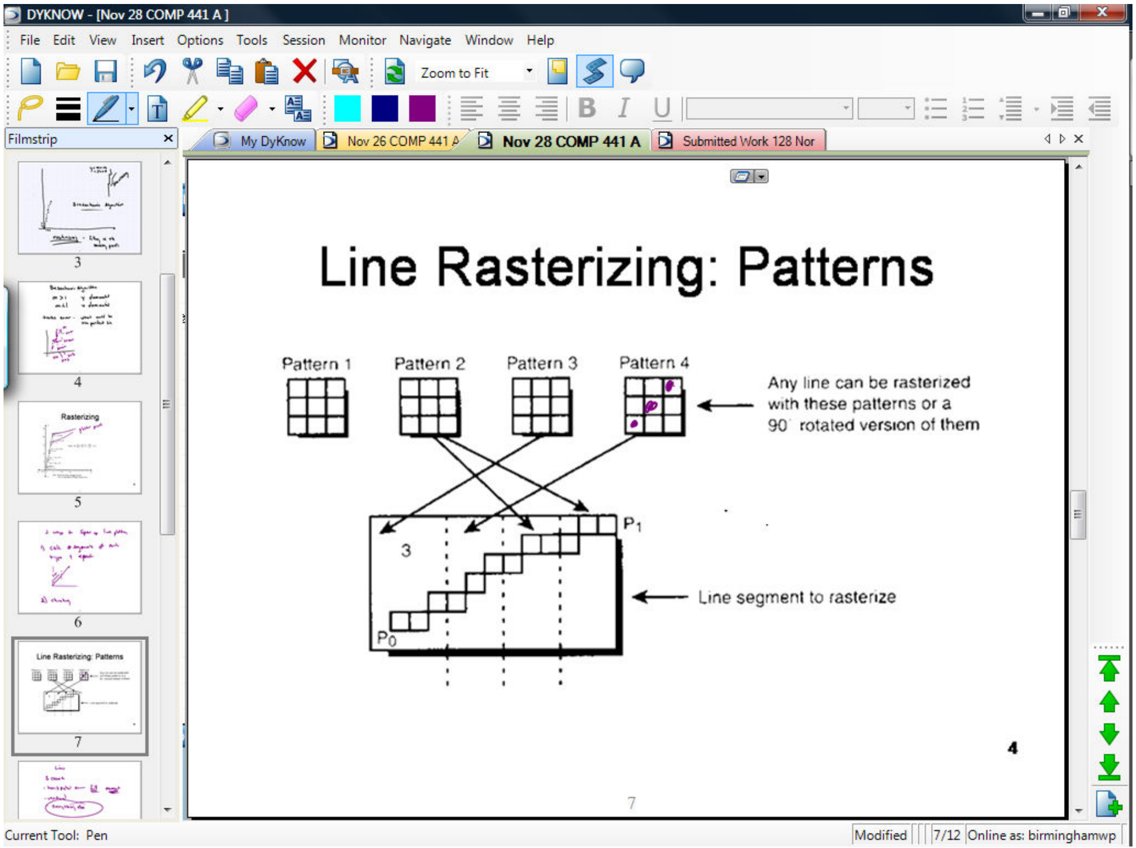

Figure 3: A PowerPoint slide.

3.1.2 Programming

One of the biggest problems with classes that involve programming is how to present and capture the development of code. The typical method used by many lecturers is to have a skeleton program that is fleshed out during the lecture. The lecturer often engages students with questions about what code should go where, what the data structures should be, etc. The students are then given the completed program at the end of the term as documentations of the lecture.

While this method can be effective, it has several drawbacks. Giving students a completed program may mask important design issues that were mentioned during the lecture. Moreover, it is virtually impossible for a student to take notes during the lecture: it is very hard to write code quickly enough to keep up with the lecturer, and typing it into an editor is hard and fraught with problems from typos.

Rather, we use CLS to capture stages of a program being created. This method works best with two screens: one screen visible to the entire class where the students can see the programming environment; and, the screen on their Tablet PC that is running the CLS. In this setup, the lecture starts with skeleton code and modifies it on the class screen, at the appropriate time, he or she "captures" a screenshot and inserts it into the CLS system as a panel such as we shown.

Since the CLS allows students their own notes, they can mark up the code as the lecture proceeds. This way, the things they find important in understanding the code are directly linked to those lines. Moreover, the lecturer can mark the code also, showing important points as coding writing progresses. We have that writing over the code is more effective than commenting, since comments can be lost in a sea of coding statements. Figure 4 shows an example of code marked up by the lecturer.

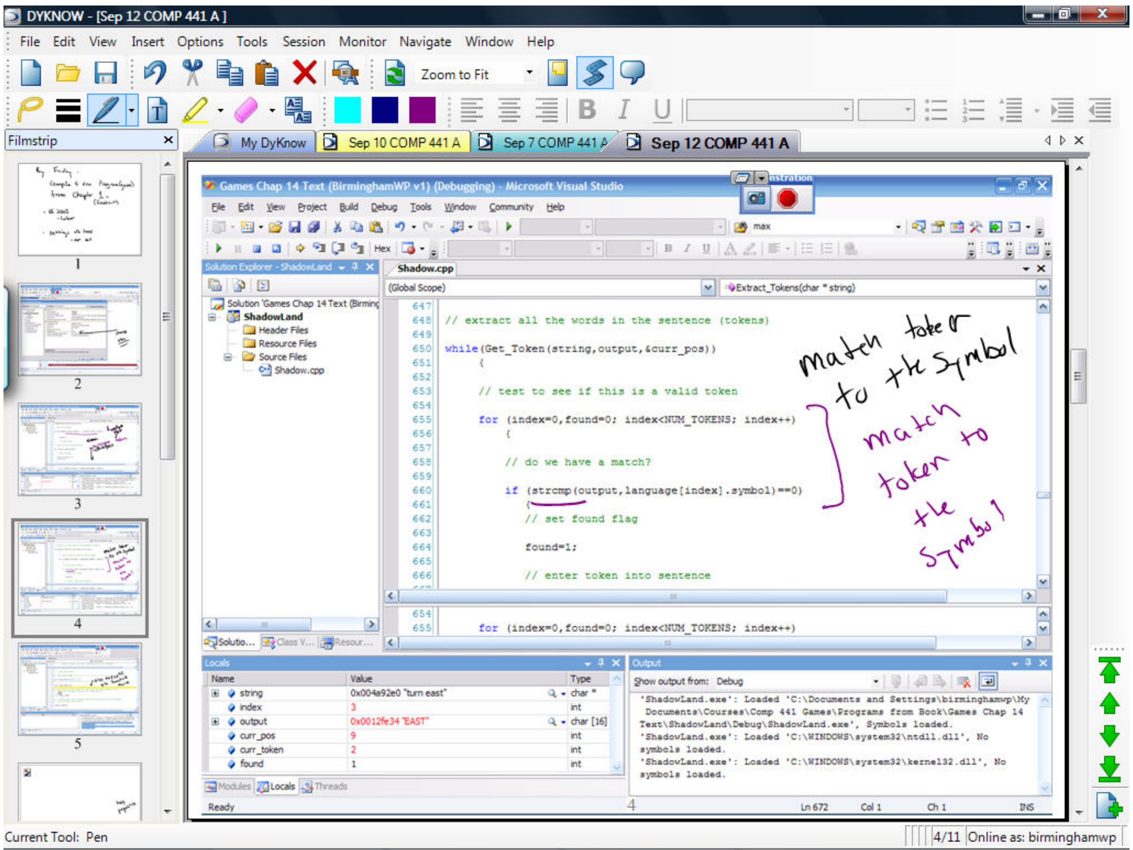

Figure 4: Marked up code.

With these examples figures, we have shown how a wide range of lecture material can be presented using classroom learning systems. The complexity of drawings, prepared diagrams, and equations is limited only by what the lecturer needs. In addition, the ability to playback pen strokes can be very useful as students study how data structures change during the algorithm execution. While we have drawn these examples from a games class, the reader can, we hope, see how they are applicable to topics across the curriculum.

3.2 Lecture: Student Perspective

In a traditional lecture (non active learning) students have commented they find themselves zoned out and not paying attention in class. In these scenarios it may not be uncommon for a instructor to lecture directly from PowerPoint without adding any additional information. In this mode students are left to quickly copy diagrams and notes as the instructor lectures and spending little time thinking about the topic at hand. The use of the classroom learning systems allows students to actively be engaged and immersed in the material. A comment from one of our surveys illustrates this: "*It's nice to be able to think about the topics as they're being written on your screen instead of spending the time writing as he's talking. Also, being able to complete an example problem and then have him review it right away helps to fix any misunderstandings about the material. Being able to save the lectures and access them from anywhere on campus since they're on the network is a HUGE advantage. DyKnow is great.*"

3.3 Lecture: Interaction

Active learning is based on involving the student during lecture. CLS systems provide many ways of doing this; in this section we describe interactive mechanisms we have found particularly effective.

3.3.1 Collecting Work

One of the most effective ways of engaging students during the lecture is to involve them in the lecture. Typically, lecturers will ask questions of students to engage students. We have found an extremely effective mechanism is to start working out an example and then have students finish it and turn it in.[8] For example, we asked students to complete the panel shown in Figure 3, and collected the results in the CLS. Figure 5 shows a student who made a common mistake.

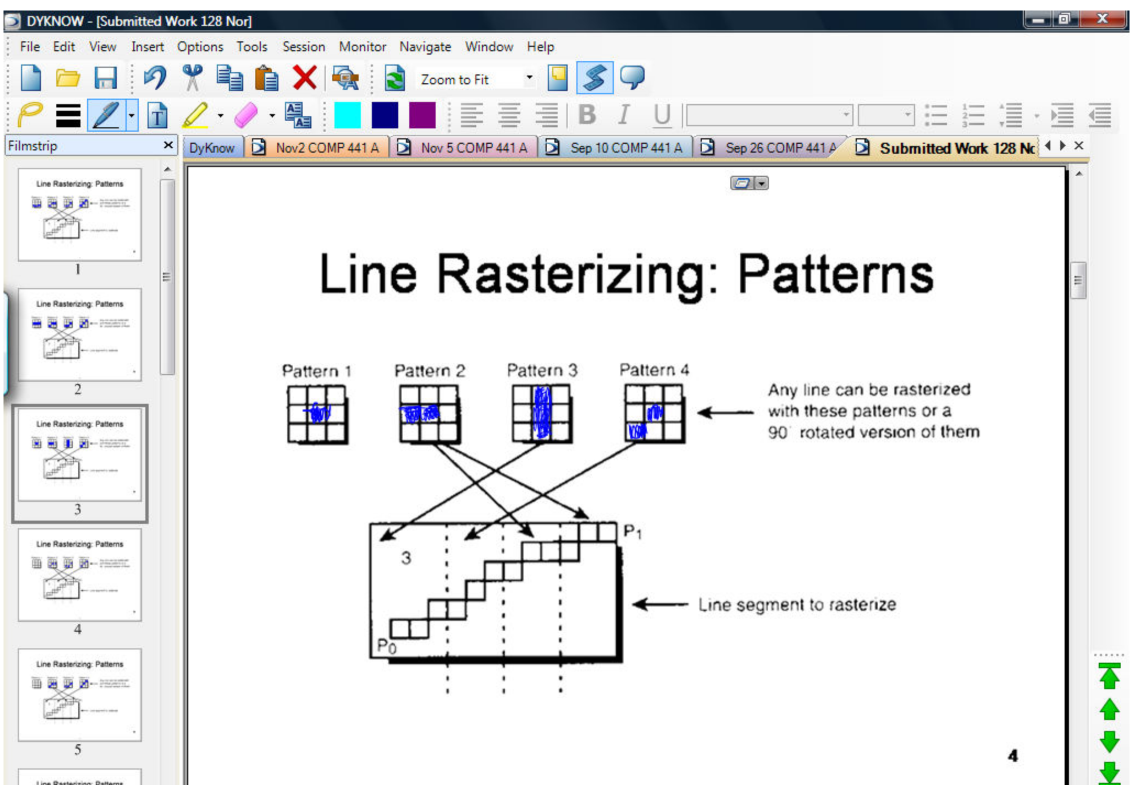

Figure 5: Student work returned during lecture.

There are several benefits to this approach. First, students will be able to immediate apply lecture concepts, thereby reinforcing those concepts or uncovering misunderstandings. Second, the lecturer can review the work as it is turned in and determine if there are common problems. For example, when we presented students with the coordinate scheme and equation in Figure 2 and asked them to plot the line, we could determine immediately that they “fell into” the trap we were hoping they would. This enabled us to effectively tie that trap into the rest of the lecture, thereby allowing the students to realize they made a mistake and how to remedy it.

3.3.2 Polling

Students are often reluctant to publically let a lecturer know they do not understand something during lecture. This can be from myriad reasons, from being embarrassed to not knowing how to phrase a question. The CLS provides a mechanism to privately let the lecturer know there is a problem. On the student machine, he or she can hit a button that indicates status: “I Understand Well”, “I Understand A Little”, or “I Do Not Understand”. The lecturer can monitor student status during lecture or ask for student status at any given time with a mouse click. If the instructor sees many students are lost, he or she can change the lecture material, add additional examples or simply ask for questions. We have found, for example, that simply saying “I see that many of you are lost: are there any questions?” breaks the ice, and students will start asking questions.

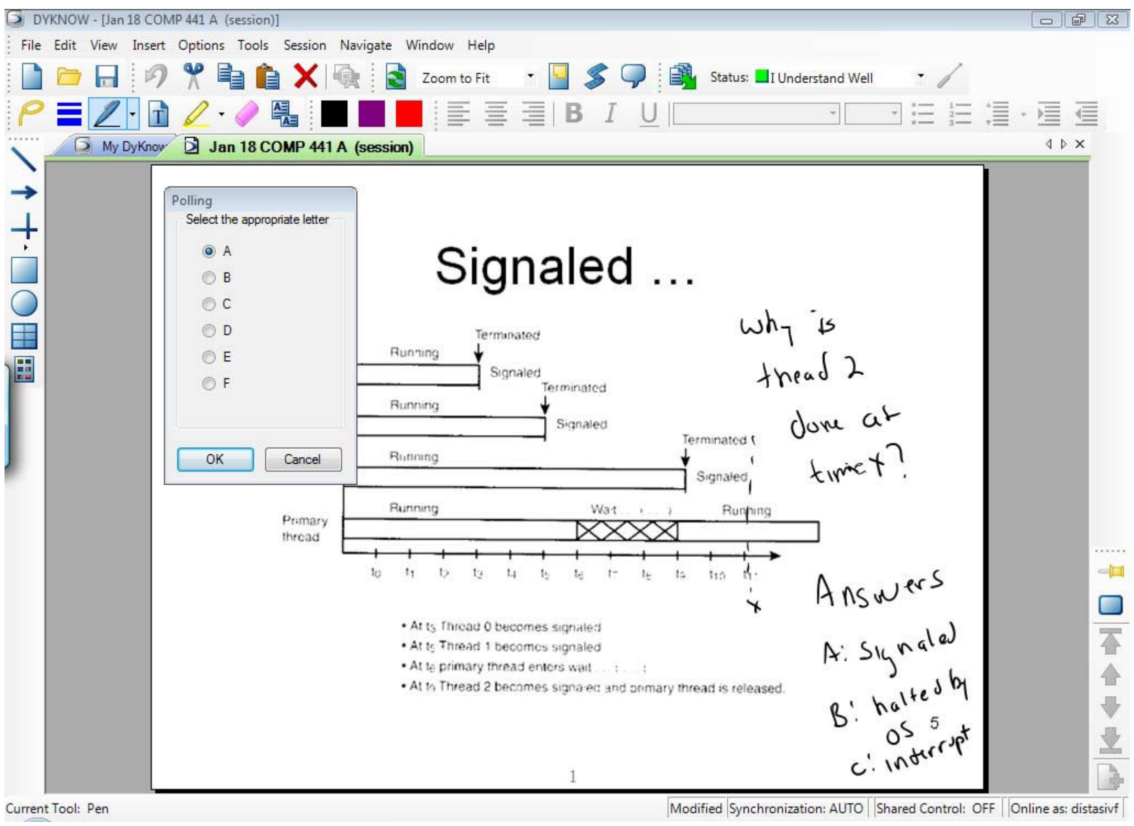

Figure 6: Example of asking a question using polling.

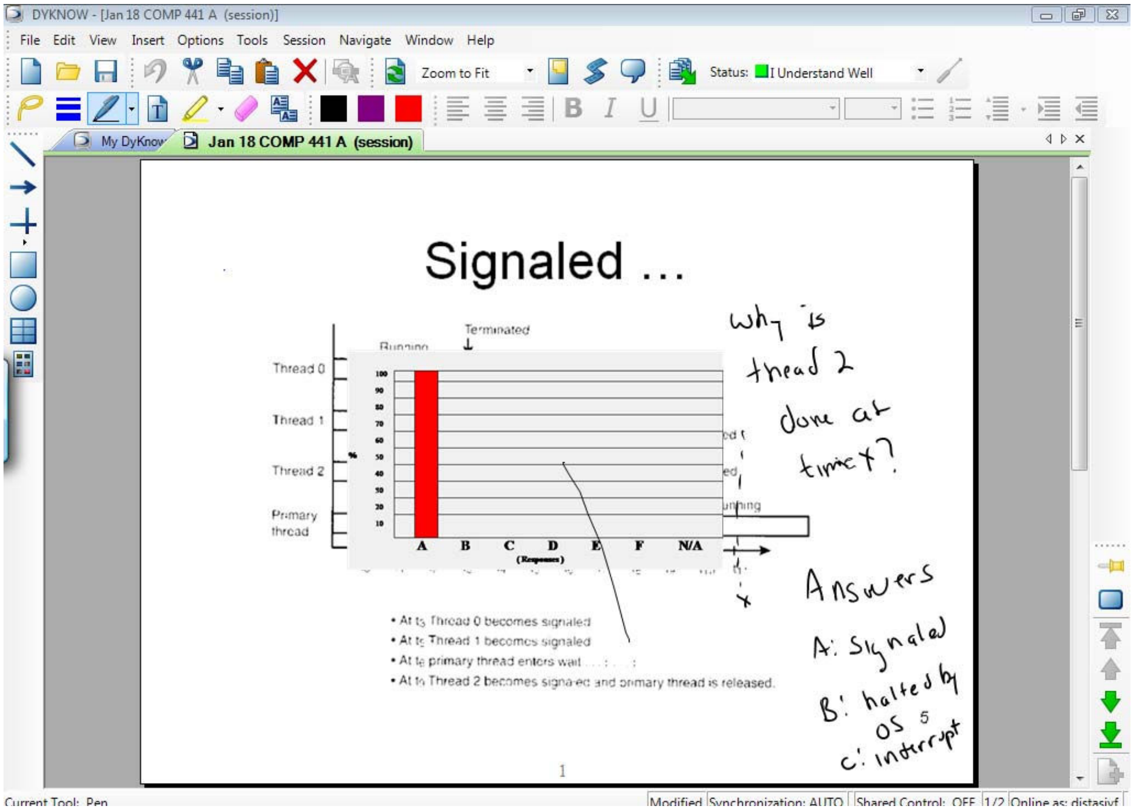

Figure 7: Example slide displaying resulting anonymous results to the class.

3.3.3 In-class Chat

As scary as it sounds, allowing students to chat during a class can be very beneficial. The CLS support chat systems. With chat, students can raise a question or make a comment without necessarily interrupting the flow of the lecture. Moreover, an eager student can answer the question, easing the burden of the lecturer. This allows for peer-to-peer help. Furthermore the chat is captured and saved so the instructor can go back to review and see if there were any key concepts brought up during the lecture that generated questions. If so the faculty member can then start the next class or send out an email, etc. to address the area and clarify the material. Likewise the instructor can also see if the chat was being abused and if so can address it during the next lecture.

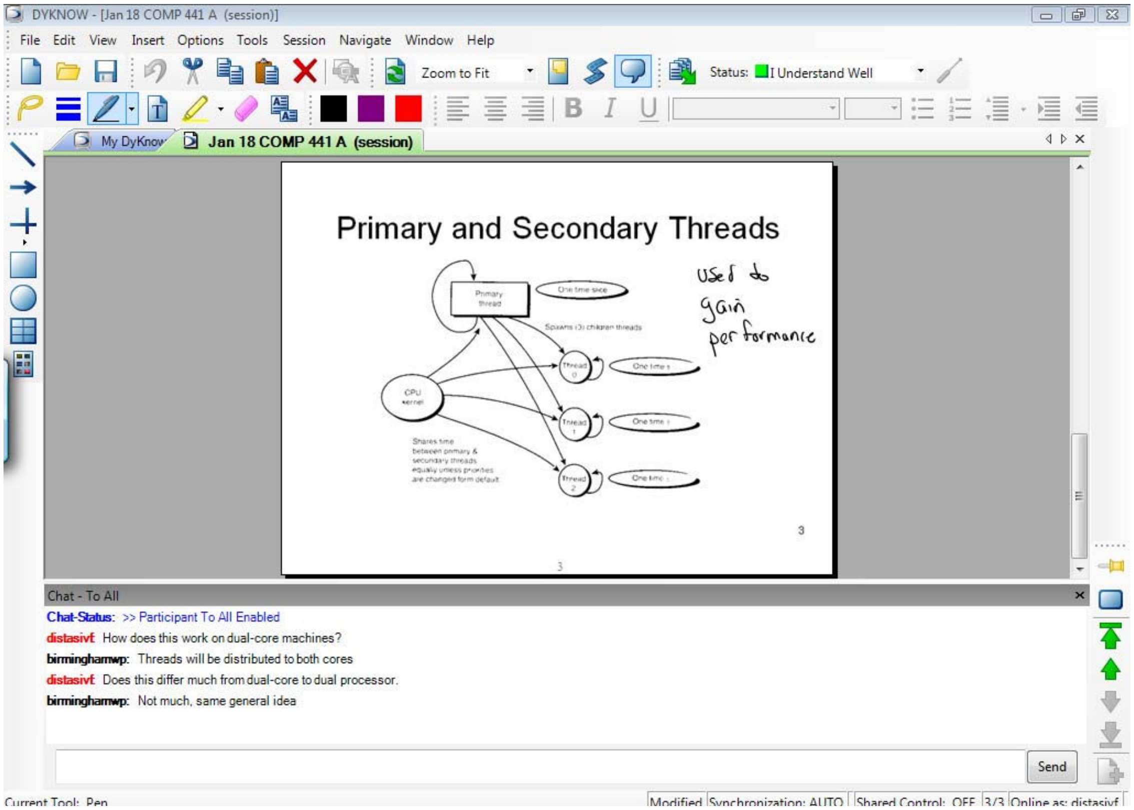

Figure 8: Example of in class chat session where students can elaborate on materials.

This feature must used with care. We have found that it can be useful, but it can also be distracting to the students and the lecturer. When the chat gets off topic, shutting it down or correcting it can be very disruptive (and sometimes ruffle the feathers of the students involved). In addition, it may be another information stream that some students find disruptive, however each individual student has the ability to show or minimize the chat window if they do not wish to participate.

4. Research Results

We are involved in a research study to better understand the efficacy of active learning technology and the attitudes of those using the technology.[11] For the past year, we have collected data from faculty and students on what they think about the technology. In this section, we present summary statistics. The data presented in this section comes only from those faculty and students using DyKnow.

Students in 13 different classes involving 16 sections offered by 10 different professors were invited to participate in an online survey about DyKnow. Both faculty and students were accustomed to PowerPoint in the classroom; hence in their responses they were essentially comparing the use of PowerPoint with the use of the projective elements of DyKnow. Students were invited to complete the survey at the end of the semester. A total of 130 students completed the survey.

Table 1 summarizes student preferences of DyKnow to other technology used in the classroom. The results indicate students have a slight preference for DyKnow over chalkboard and overhead transparencies. OneNote or PowerPoint are slightly preferred to DyKnow. Through interviews and comments, it appears that OneNote is preferred because it is perceived to be easier to use and faster to load files. We have observed students using both programs together in the classroom: DyKnow to capture lecturer notes and OneNote is used for student note taking. The preference for PowerPoint is a bit confusing, as it has about the same performance as DyKnow and has more overhead in terms of loading and saving files and does not have the benefit of shared note taking, however familiarity with PowerPoint may have a significant impact on this observation.

Table 1: Student satisfaction in comparison to other technology.

Question	(N = 154) Mean and sd
*lecture and chalkboard?	*M* = 4.18, *sd* = 1.825
*overhead transparencies?	*M* = 4.50, *sd* = 1.629
*PowerPoint or OneNote?	*M* = 3.71, *sd* = 1.633

1 = DyKnow provides a very significant disadvantage.
4 = DyKnow provides neither and advantage nor a disadvantage.
7 = DyKnow provides a very significant advantage.

Student's view of the usefulness of DyKnow in helping their performance is given in Table 2. We observe a picture that is neutral to slightly negative. In general the students do not perceive that DyKnow, and perhaps active learning, provides them and advantage or disadvantage over traditional lecture tools. We believe that part of this is because DyKnow and active learning are new, and there is unfamiliarity with the application. In addition, other data indicates strongly that if a faculty member is not well-versed in DyKnow, it has a strong negative impact on student acceptance. There were faculty who self reported that they were not facile with the technology.

Table 2: Student ratings of usefulness.

Question	(N = 154) Mean & sd
Using DyKnow enhances my understanding of course material.	*M* = 3.12, *sd* = 1.88
Using DyKnow provides me with a better set of notes.	*M* = 3.53, *sd* = 2.09
I am more attentive during class because DyKnow is used.	*M* = 2.70, *sd* = 1.67
I am more confident during exams when I have studied material from my DyKnow notes as compared to when I have studied from traditional paper notes.	*M* = 2.84, *sd* = 1.74

1 = Strongly Disagree
4 = Neither Disagree or Agree
7 = Strongly Agree

Table 3 show a very different picture of the technology than does Table 1: instructors *strongly* prefer DyKnow to other types of technology. The difference between students and faculty perceptions of satisfaction require more study to understand fully, but we suspect that DyKnow offers workload reduction (fewer files to move around), the ability to present and interact with material easily, better feedback to the instructor on student understanding during lectures, and a record of what transpired during the lecture. All of these offer, in total, a significant advantage over existing technology.

Table 3: Instructor satisfaction in comparison to other technology.

Question	(N = 21) Mean and sd	Students, (N = 154) Mean and sd
*lecture and chalkboard?	$M = 6.14$, $sd = 0.910$	$M = 4.18$, $sd = 1.825$
*overhead transparencies?	$M = 6.29$, $sd = 0.717$	$M = 4.50$, $sd = 1.629$
*PowerPoint or OneNote?	$M = 5.57$, $sd = 0.978$	$M = 3.71$, $sd = 1.633$

1 = DyKnow provides a very significant disadvantage.

4 = DyKnow provides neither and advantage nor a disadvantage.

7 = DyKnow provides a very significant advantage.

5. Discussion

The data are clear in distinguishing satisfaction levels between faculty and students. The strong faculty acceptance of the technology we believe will eventually work its way to student acceptance. In addition, as students learn to use the technology, we believe that will begin to find it more useful. We have noticed that freshmen and sophomores seem to have a higher satisfaction level than upper classmen. This may be because their study habits and lecture behaviors are not as deeply formed as with upper classmen. Also, DyKnow and active learning were introduced into classes fairly recently, and were a disruption to the upper classmen. The sophomores and freshmen do not know anything else, so to them it is the status quo.

Similar to other surveys conducted at other institutions our results show some differences between faculty and student perceptions of using these tools. However our student satisfaction does not seem to be as high as others have reported in the literature. [9, 10]

We are continuing our longitudinal studies, as we are tracking both use and satisfaction levels over several semesters.

6. Summary and Conclusion

Classroom learning systems combined with Tablet PCs are an effective way of employing active learning in the CS curriculum. We use CLS in classes ranging from freshmen programming through our upper-division courses. The ability to deliver notes directly from the instructor to the student permits reduces faculty workload and provides a way for the instructor to deliver information that was virtually impossible to deliver before. For example, the ability to "playback" pen strokes allows students to see how data structures are changed. The ability to capture and annotate code as it is being written greatly simplifies—in fact, makes feasible—student note taking during coding lectures.

Active learning is facilitated by students reporting their status privately. This is an easy way to inform the instructor when he or she does not understand a concept. The ability to start an example and have students finish and return it during lecture engages students in the topic and allows the instructor to accurately and in real time, gauge student understanding.

The survey results of satisfaction and utility are mixed. Students have mixed feelings, preferring the CLS system to overheads and chalkboards, but not to PowerPoint and OneNote. Nor do they see an advantage. Faculty, on the other hand, greatly prefer CLS to other technologies. We believe that as students become facile with the technology and its use is the norm, their satisfaction ratings will increase. We are conducting further studies to see if this is, in fact, the case.

Acknowledgements

We thank our research colleagues Gary Welton, Ananda Gunawardena, and Dave Ayers. The authors thank Microsoft Research for providing partial funding to conduct this research through a 2006 Microsoft Research Tablet-Based Computing Initiative Grant. We would also like to thank Hewlett-Packard Corporation for their continued and valued partnership; DyKnow for their valued partnership and support of the project through additional software licensing and listening to our feedback; and Dr. Jane Chu Prey from Microsoft Research, External Research and Programs. We would also like to thank our students and faculty who have participated in pushing the future use of new educational technologies and for their continued feedback, in particular: Stacy Birmingham, Paul Kemeny, Eric Anderson, Dorian Yeager, Blair Allison, Christiaan Gribble, Fred Jenny, and Susan Cramer.

Bibliography

1. *Improving learning in CS1 via Tablet-PC-based in-class assessment.* Kolie, K. and Singer, D. Canterbury, United Kingdom : ACM Press, New York, NY, 2006. Proceedings of the 2006 International Workshop on Computing Education Research. pp. 119-126.

2. *Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses.* Anderson, Richard, et al. Norfolk, VA : ACM Press, New York, NY, 2004. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education. pp. 56-60.

3. *Multimodal Communication in the Classroom: What does it mean for us?* Denning, Tamara, Griswold, William G. and Simon, Beth. Houston, TX : ACM, 2006. Proceedings from SIGCSE. pp. 219-223. 1-59593-259-3/06/0003.

4. *Teaching Courses with Tablet PC: Experience and Student Feedback.* Biswas, Saroj. s.l. : ACM, 2007.

5. *Using Pen-Based Computers Across the Computer Science Curriculum.* Berque, Dave, Bonebright, Terri and Withesell, Michael.

6. *An Evaluation of a Broad Deployment of DyKnow Software to Support Note Taking and Interaction Using Pen-Based Computers.* Berque, Dave. s.l. : Consortium for Computing Sciences in Colleges, 2006. pp. 204-216.

7. *Tablet PC - Is It Worth It? A Preliminary Comparision of Several Approaches to Using Tablet PC in An Engineering Classroom.* Lord, Susan and Perry, Leonard. s.l. : American Society for Engineering Education, 2006.

8. Software, Dyknow. *www.dyknow.com.* [Online]

9. Presenter, Classroom. www.cs.washington.edu/education/dl/presenter. [Online]

10. XP, Conference. research.microsoft.com/conferencexp. [Online]

11. *Evaluating learning software in the classroom: a preliminary study.* DiStasi, Vincent F., et al. [ed.] Jane C. Prey, Robert H. Reed and Dave A. Berque. West Lafayette, IN : Purdue University Press, 2007. The Impact of Tablet PCs and Pen-based Technology on Education. pp. 43-50. 978-1-55753-461-3.