

# **AN EVALUATION OF A BROAD DEPLOYMENT OF DYKNOW SOFTWARE TO SUPPORT NOTE TAKING AND INTERACTION USING PEN-BASED COMPUTERS**

**Dave Berque**  
**Computer Science Department**  
**DePauw University**  
**Greencastle, IN 46135**  
**(765) 658-4735**  
***dberque@depauw.edu***

## **ABSTRACT**

This paper describes a department-wide deployment of DyKnow software to support collaborative note taking; classroom interaction; out of class review, replay, and grading of classroom materials; and computer monitoring. We describe diverse ways the software can be used to support computer science education and present the varied hardware environments (including standard laptops, Tablet PCs, interactive pen displays, and graphics tablets) that have supported the deployment. Results from surveys administered to 10 teachers and 81 students are presented. Collectively, these students have taken 431 classes using DyKnow tools. The survey results demonstrate the value students and teachers place on specific aspects of our approach and suggest areas for improvement.

## **1. INTRODUCTION AND SIGNIFICANCE**

Computer Science instructors are increasingly exploring the use of Tablet PCs (essentially pen-enabled laptops) and other pen-based computing devices to support computer science education. Projects recently reported in the literature include the use of these devices to support instructor presentations, student submissions, grading, and e-text reading. The interested reader is referred to [1, 4, 5, 6] for a sampling of recent experimental deployments.

The author's own previous work has included the design, use, and evaluation of a system that supports electronic student note taking and in-class interaction. The author reported on this work in [2] and extends this work here.

Several developments motivate the current paper and distinguish it from the prior literature. The earlier software systems described by the author have been greatly expanded and are now available as an integrated pair of software tools named DyKnow Vision and DyKnow Monitor (patent pending) [3]. The expanded tools support several pedagogical approaches that have not been previously described. In addition, the new tools are now being used broadly across teaching styles, courses, disciplines, and grade levels. DyKnow tools have been used to support courses in disciplines including mathematics, history, economics, Japanese language, English, and (as described in the remainder of this paper) computer science. With respect to computer science education the tools have been used in a variety of environments including high schools, junior colleges, liberal arts colleges, and larger universities.

At the author's own school, essentially every course taught in the computer science department uses DyKnow tools. For example, during the past three semesters

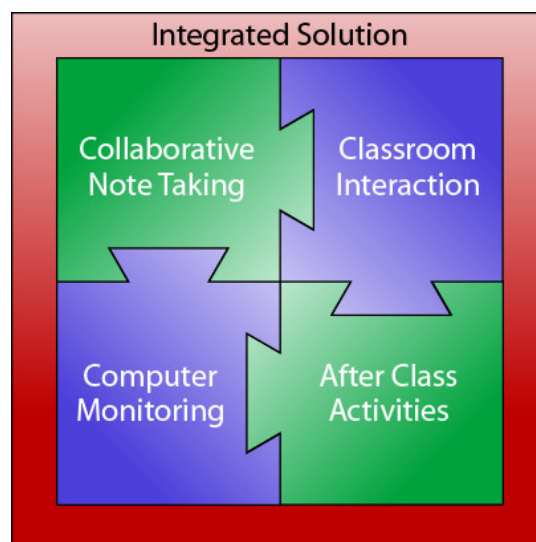
alone 43 computer science courses (all except senior project) used the system. This broad usage makes it possible to report on the use of, and reaction to, the system across different courses. While many papers in the computer science education literature report on the use of new technologies or pedagogies as deployed by a single instructor in a single course, few report on evaluations of broader deployments that span multiple courses and instructors as this paper does.

As described in **Section 3**, DyKnow deployments have been supported by varied hardware devices ranging from standard laptops and laboratory machines to pen-based computing devices including Tablet PCs. Part of our evaluation seeks to understand the degree to which students and teachers are making use of pen-based input when given the choice between using a pen and using a keyboard. This information may be used to inform decisions other schools are making regarding the role, if any, of pen-based computing in their environments.

In the remainder of this paper we provide an overview of DyKnow teaching tools with emphasis on showing how the tools have been used to teach computer science. We then briefly summarize the hardware environments that have been used in conjunction with DyKnow software. Next, we present results of a survey of 81 students who have collectively taken 431 classes with DyKnow tools and a second survey of 10 teachers who have collectively taught 90 courses with the system. Finally, we outline lessons learned and suggest areas for future work.

## 2. SUPPORTED PEDAGOGIES

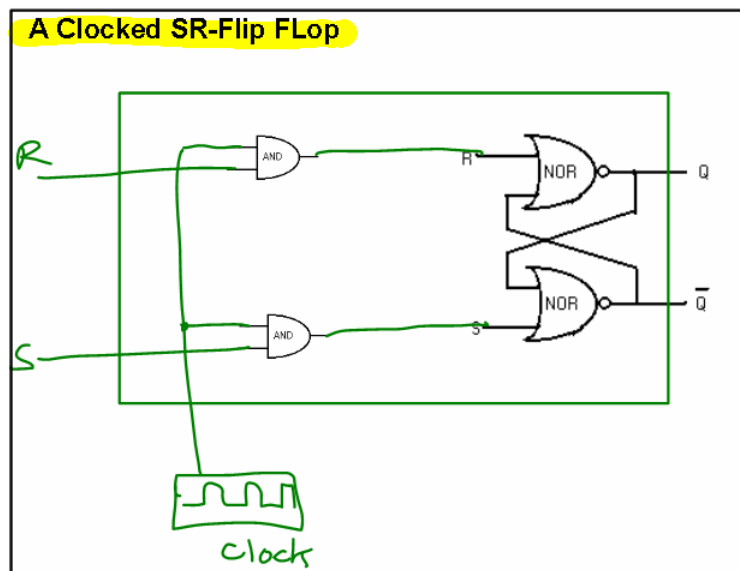
As shown in **Figure 1** the DyKnow system supports teaching and learning by facilitating four mutually supportive activities: collaborative note taking; classroom interaction; out of class review, replay, and grading of classroom materials; and computer monitoring. In the remainder of this section we explain key aspects of the system largely by presenting usage examples from a variety of computer science courses. Understanding the variety of ways the system is used and appreciating how the separate usage patterns integrate together sets the stage for the evaluation presented in **Section 4**.



**Figure 1. Mutually Supportive Benefits of DyKnow Software**

## 2. 1 Collaborative Note Taking

DyKnow software helps students develop a multi-page electronic notebook for each class period. We refer to this notebook as a "shared artifact" because it includes content provided by the teacher supplemented with private student annotations. During class the teacher can extemporaneously draw sketches directly on a pen-based computer such as a Tablet PC. The teacher can also use a keyboard to type material, and can import material (including *PowerPoint* slides, live web pages, and screen shots from other applications) for use during class. Everything that the teacher writes, types, or imports is immediately displayed on each student's machine. For example, **Figure 2** shows a screenshot taken from DyKnow that includes typed text, highlighting, prepared graphics (the gates and flip-flop are available as graphics items on DyKnow's palette tool) and free hand drawing. Some of this content was prepared prior to class and additional content, for example the sketch of the clock pulse, was added by the teacher during class. All of this content was interactively transmitted to each student's display.

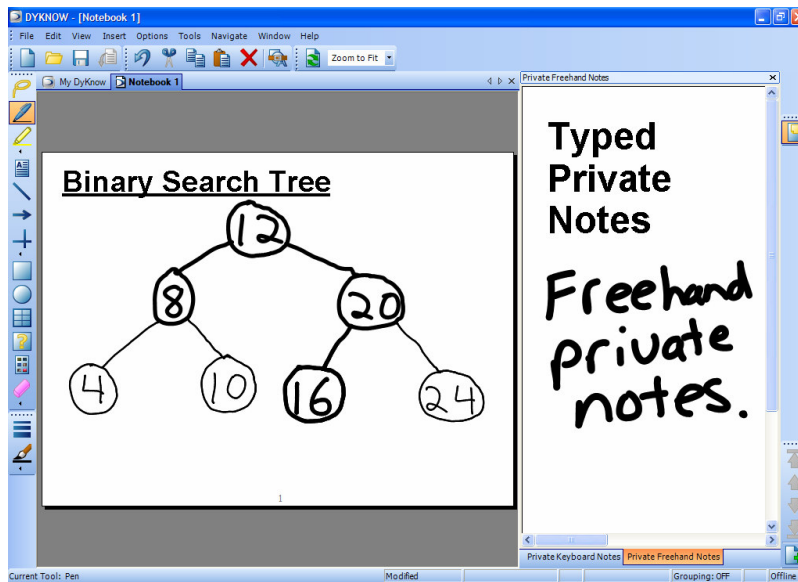


**Figure 2. Various Content Types Available in DyKnow**

Each student can personalize his or her electronic notebook by making private annotations (typed or freehand) to supplement the teacher's material. Student annotations can be made directly on top of the teacher's content or in a private notes "margin" that is associated with each page of the notebook. For example, the teacher could use DyKnow to display a sketch of a binary search tree. The teacher could then extemporaneously insert a value into the tree while explaining the associated algorithm. The original tree and newly added value would be displayed on each student's computer. The teacher could then ask the students to add several additional values to the tree; each student could add these values directly to the diagram that the teacher had supplied.

On the other hand, if the teacher orally defined tree terminology such as "root," "parent," and "sibling," the students might choose to write or type these definitions in their private notes margin. **Figure 3** shows the DyKnow interface. The thick ink in the left-hand pane was drawn by the teacher and the thinner ink was added by a student. The right pane shows the student's private note area. As explained more fully in

section 2.4, at the end of class students can print or save their notebook for later review and study.



**Figure 3. DyKnow Collaborative Note Taking Interface**

DyKnow teachers often share prepared content with students following a "pause, practice, and disclose" pattern that we call progressive disclosure. This technique encourages students to contribute to the shared artifact notebook, thereby helping them to think about content as it is being presented. For example, a teacher might provide the class with a DyKnow page that poses a short programming problem. The teacher would pause to give students time to type or write a code fragment to solve the problem. Only after the students had practiced writing the code fragment, would the teacher disclose a second DyKnow page containing a model solution.

## 2. 2 Classroom Interaction

The National Research Council, urges educators to provide "active learning environments for all students, even in large section, lecture-dominated courses" [7]. As described below, DyKnow has numerous features that support active learning. Moreover, the results of the interactive activities can be included in each student's collaborative notebook.

**Polling:** DyKnow's polling tool allows instructors to present a multiple choice poll to the class. These polls can be used to gauge mastery of content (does the loop iterate 2, 4, 8 or 16 times) or to give students a chance to anonymously indicate their level of understanding of a topic (do you understand recursion very well, well, slightly, or not at all)? The results of these polls can be displayed as charts or graphs and can optionally be pasted into the DyKnow shared artifact notebook.

**Student Submissions:** DyKnow's student submission feature allows students to share complex work with the teacher or with the class. This process can be initiated by the students (e.g., a voluntary submission to an in-class problem) or by the teacher (e.g., when collecting a quiz). For example, the instructor might ask students to write a Java function meeting some specification. The students might work alone or in groups as they type or write their solutions in their DyKnow notebook. Work can be

submitted by one or more students; the instructor may optionally allow the submissions to be anonymous.

Submissions can be viewed privately by the instructor, or projected to the front of the room for the entire class to see. Alternatively, the instructor can add one or more submissions to the collaborative notebook. This latter approach allows students to leave class with several different solutions to a problem. Introductory programming students have often been amazed to learn that there is more than one correct way to write a function.

Student submissions can contain a wide range of content including screen shots taken from other applications. For example, students in a computer organization class can use a digital logic simulator to construct a circuit. The students could then paste an image of this circuit into a DyKnow notebook. This content could then be submitted and annotated just like any other DyKnow content.

**Student Control:** Student control allows students to share work with the class on a stroke-by-stroke basis. For example, after explaining the standard Selection Sort algorithm, a teacher can draw an array containing ten integers. Next, the teacher can give control to a student volunteer. As the student redraws the array to show its state after the first pass of Selection Sort completes, her pen strokes are transmitted to the class as they are drawn. Multiple students can also be given simultaneous control, allowing them to collaborate in a shared work area.

**Embedded Web Pages:** Live web pages can be embedded in a DyKnow notebook and transmitted to students along with other DyKnow content. For example, a teacher can embed a web page that contains an interactive animation of several sorting algorithms. Students can use the animation to count the number of data moves that are required to sort various sized arrays. As the students interact with the simulation they can record their results in a table that is part of the DyKnow notebook. Thus, the live web page and associated table become part of each student's collaborative note book. Web pages can also be captured as images and annotated along with other DyKnow content.

**Demonstration Mode:** This feature allows applications to be demonstrated to the entire class. For example, in a Human Computer Interaction class, students might be asked to prototype a voting machine interface. After the prototypes have been built students can explain their design decisions. While a student is offering an explanation, the teacher can share the student's screen with the class. This allows the student to interactively demonstrate the prototype for the class to see. Snap shots of the demonstration can also be inserted into the collaborative notebook thereby providing a record for annotation and later study.

### 2.3 Computer Monitoring

DyKnow supports computer monitoring features that instructors can use to gauge student progress and focus student attention. For example, if students are using a development environment to write a program during class, the teacher can see a live thumbnail (or larger image) of each student's display. This allows the teacher to see if individual students are on track. As is the case with all other DyKnow content, these images can be pasted into a DyKnow notebook where they can be annotated easily.

Additional monitoring functionality allows the teacher to send messages to individual students, groups of students, or to the entire class. In addition, the teacher can blank student screens and can lock their mice and keyboards. Finally, the teacher

can focus attention by limiting the applications a student can run during class. For example, a teacher may allow students to run only *DyKnow* and *BlueJ*. Alternatively, a teacher may allow students to run any application except those on a specified list.

## 2.4 Review, Replay, and Grading

At the end of each class students can print their notes (to paper or to a PDF file), or they can save their notes electronically. Saved notes can be stored to disk or to the DyKnow server. Notes saved to the server are organized by semester, course, and date. Thus, a student can retrieve notes from any semester for review and study from any Windows computer with Internet access. Some of our students and faculty members also run the software on a Mac or Linux computer running an appropriate emulator such as Virtual PC. Retrieved notes can also be replayed stroke-by-stroke. This can show, for example, how a pointer based stack changes as items are pushed and popped.

While the DyKnow server is a licensed product, the client software is a free download from [www.dyknow.com](http://www.dyknow.com). Thus, students can freely install DyKnow on their Windows-based desktop, laptop, or Tablet PC. Some students have also used DyKnow on MacIntosh or Linux machines running a suitable emulator such as Virtual PC.

DyKnow notes can also be reviewed and annotated outside of class by instructors. For example, after class, instructors frequently add comments and/or grades to quizzes collected with DyKnow during class. The graded quizzes can then be returned electronically to the students.

## 2.5 Integration of Features

It is important to note that the usage patterns described in 2.1 – 2.4 can be combined in a variety of ways to provide a complete class session. The remainder of this subsection demonstrates this type of integration through a scenario that was first reported in a simplified form in [2].

**Scenario:** After explaining the standard Selection Sort algorithm to an introductory computer science class, the teacher draws a ten element array and fills it with values. As the teacher draws, the diagram is transmitted to the students stroke by stroke. The teacher demonstrates the first pass of the algorithm by circling the largest element in the array and swapping it into its correct position (which involves erasing and rewriting values). Again, as the instructor makes each change it is reflected on each student's display interactively.

As the instructor continues to explain the algorithm he verbally notes that the algorithm sometimes swaps an item with itself. Several of the students add a private annotation to their notebooks to help them remember this potentially puzzling point. Since each student's annotations are private, they are not seen by anyone else in the class.

Next, the instructor asks for a volunteer to carry out the next pass of the algorithm. Sandy volunteers and the teacher gives her control of the system. Now that she has control, although Sandy is still seated at her desk, everything she draws is *immediately* transmitted to every other student. Sandy circles the largest element in the unsorted portion of the array and swaps it into its correct location. At this point the

teacher takes back control, before turning control over to another student to execute the next pass of the algorithm.

Convinced that the students understand the mechanics of the algorithm after several students have correctly executed a pass, the teacher initiates a multiple choice poll and asks the students whether they believe the algorithm runs in time  $O(N)$ ,  $O(N^2)$ ,  $O(\log N)$ , or  $O(N \log N)$ . The results of the poll indicate that only 25% of the class understand that the algorithm runs in time  $O(N^2)$ . Because of this, the teacher adds a blank page to the notebook and draws some sketches on the page to show how to count the number of comparisons made by the algorithm, thereby establishing its running time.

Next the teacher launches an integrated programming environment and opens a file containing a partially written Selection Sort program. The teacher uses DyKnow's demonstration mode to project the program onto the student displays and demonstrates that although the program compile and runs, it fails to sort its input because a key portion of the code has not been completed. The students watch the teacher's live demonstration on their screens and ask questions about the portion of code that the teacher has asked them to write. Next the students start their own copies of the integrated programming environment and work in groups to complete and test the copy of the program provided by the teacher. As the students work, they toggle back and forth between their DyKnow notebooks (containing the outline of the Selection Sort algorithm) and their programming environments (containing the skeletal code).

While the students work, the teacher uses DyKnow's monitoring features to watch their progress from his workstation. Noticing that one group is off track, the teacher walks over to them and asks if they need help. It turns out that the students are stuck on a routine syntax error that the teacher is able to address easily.

After most of the groups have completed their work, the teacher asks for volunteers to share their work with the class. One group pastes their solution into DyKnow from the programming environment. This group then uses DyKnow's student submission feature to share their solution with the teacher who makes it available for the class to see. As the teacher discusses the answer he adds some annotations and highlights to point out some important lines of code. These are transmitted to the entire class.

At this point a student in another group says "we did something almost the same, but ours does not work." Can you tell us why? The teacher invites this group to submit their work, and the class works together to identify the small bug in the code.

Several weeks later, while studying for an exam, students are reviewing the Selection Sort algorithm in their electronic notebooks. After reviewing the pages that contain the instructor's description of the algorithm, one student decides to replay the in-class exercise stroke by stroke. As the student watches, the elements of the array are redrawn, showing how the data elements move to their new locations.

### **3. SUPPORTING HARDWARE**

This section summarizes the pen-based computing hardware that supported our deployment in order to set the context for the evaluation in **Section 4**. Currently the most popular type of pen-based computing device is the *Tablet PC* (a pen-enabled laptop). A related device is the *interactive pen display* which can be attached to a standard desktop computer in place of a traditional monitor. Interactive pen displays

are essentially LCD displays that the user can write on with a special pen. Finally, a *graphics tablet* is an inexpensive (often less than \$100) USB device that attaches to a standard laptop or desktop computer. The user can draw on the graphics tablet with a special pen but must look at the computer display to see the drawn content.

DyKnow teaching tools run on all of the hardware devices described above, as well as on standard laptops and desktops. At DePauw we have used DyKnow primarily with interactive pen-displays such as those shown in **Figure 4**; however, we have also used the system with Tablet PCs and with laptop computers (sometimes supplemented with graphics tablets).



**Figure 4. Interactive Pen Display**

## **4. EVALUTION**

### **4.1 Survey Background**

At the end of the spring 2005 semester all DePauw University computer science majors and minors were invited to participate in an anonymous survey related to their use of DyKnow tools. Of the 120 students who were invited to participate, 81 students completed the survey. Sixty-four (79%) respondents were male and 17 (21%) were female. The survey consisted of more than 30 questions, many with multiple parts. Students were compensated with a gift certificate that could be used toward the purchase of food at a popular local restaurant.

The student respondents had taken a total of 399 computer science courses using DyKnow as compared to only 78 computer science courses without DyKnow. This latter set of courses primarily consists of courses taken by seniors during their first year at DePauw (just prior to a broad DyKnow deployment). The participants had also collectively enrolled in 21 economics courses, six English courses, four Japanese courses, and one communications course using the system. Thus, the total number of enrollments in DyKnow courses was 431.

A survey was also administered to faculty members who had taught with the system. Because a reasonably large number of computer science students had also taken economics courses using DyKnow, we surveyed both computer science and economics instructors. Surveys were sent to ten faculty members and all ten surveys



were returned. Six surveys were from computer scientists and four from economists. Collectively these instructors had taught 90 courses using the DyKnow software.

#### 4.2 Overall Value and Impact on Pedagogy

The broadest survey questions explored the extent to which students and teachers valued their use of the DyKnow system. For example, the students were asked to indicate their level of agreement with the statement: *Overall DyKnow has had a positive impact on what I have learned as a Computer Science major or minor.* Fifty nine (73%) of the students strongly agreed with this statement, 20 (25%) agreed somewhat, one student was neutral, one student disagreed somewhat, and no students strongly disagreed.

In addition 60 (74%) of the students answered "yes" when asked if they had ever wished DyKnow was used in a course that did not use it. Conversely only five (6%) of the students answered "yes" when asked if they had ever wished DyKnow were not used in a course that did use it.

When broad questions were posed to teachers similar results were obtained. For example, 100% of teachers agreed or strongly agreed with the following statements:

- Overall, DyKnow has had a positive impact on my experience as a teacher.
- DyKnow has had a positive impact on student learning in my classes.

Additional questions probed more deeply in order to elucidate *exactly why* teachers and students value their use of DyKnow. Each group was asked to specify whether they placed *no value*, *small value*, *moderate value*, or *significant value* in a variety of areas. Table 1 summarizes aspects of DyKnow that teachers and students both find to be beneficial. The given values show the percent of teachers (T) and students (S) who rated each item with *moderate value* or *significant value*.

Item Rated	T	S
Enhancing student understanding of material and concepts as they are presented during class.	90%	95%
Enhancing student understanding of material and concepts while students are studying outside of class.	100%	87%
Providing an accurate set of notes.	100%	100%
Allowing students to find all their notes in one place.	90%	97%
Allowing students to add their own notes to the teacher's content.	100%	98%
Doing in-class exercises to practice with content.	100%	93%

**Table 1. Items both teachers and students find to be valuable.**

Surprisingly, there were not many places in this section of the survey where student and teacher responses varied greatly. The questions that exhibited the most notable differences are shown in Table 2.

Item Rated	T	S
Allowing students to see how their peers have solved problems during class.	100%	71%
Allowing students to submit work anonymously.	80%	62%

**Table 2. Items that teachers and students disagreed on.**

Although fewer students value the ability to submit work anonymously, this feature may be extremely important for those students who do value it.

Several additional rating items were presented to teachers. Key items, along with the percentage of teachers who rated the item as having *moderate value* or *significant value* are shown in Table 3.

Item Rated	T
Allowing teachers to see how students have solved problems during class.	100%
Providing teachers with a record of what they have done during class.	100%
Allowing teachers to share class notes with other teachers.	50%

**Table 3. Additional questions posed to teachers.**

Teachers were also asked to indicate their level of agreement with the following statements. The values in Table 4 show the percentage of teachers who rated the item with *agree somewhat* or *strongly agree*.

Item Rated	T
I can have students do in-class problems more easily when DyKnow is used.	90%
DyKnow enhances my ability to use pedagogies that I already valued.	80%
I cover more material when DyKnow is used.	60%

**Table 4. Additional questions posed to teachers.**

### 4.3 Frequency of Specific Usage Patterns

Teachers were also asked if they used specific DyKnow features *never*, *rarely*, *sometimes*, or *frequently*. The most commonly used features are listed in Table 5. The given values show the percentage of teachers who indicated that they *sometimes* or *frequently* use the indicated feature.

Item Rated	T
Add emphasis marks (underlining, circling, highlighting) to prepared notes during class.	100%
Expect students to annotate notes in class.	100%
Use DyKnow to transmit problems or exercises for students to complete.	100%
Review DyKnow notes while writing an exam or quiz.	100%

Use the DyKnow "screen grab" feature to import content from other applications into DyKnow for use during class.	90%
Adjust the pace or content of class based on student work viewed through DyKnow.	70%
Use DyKnow authoring tools to prepare content.	70%

**Table 5. Frequently used features.**

Finally, the features least frequently used by teachers are listed in Table 6.

Item Rated	T
Use DyKnow Monitor.	20%
Import a PowerPoint slide show to use in DyKnow for use during class.	20%

**Table 6. Infrequently used features.**

Returning to **Figure 1** we note that teachers are making moderate to heavy use of three of the four core aspects of the system. Computer monitoring, however, is being used much less.

#### **4.4 Keyboard versus Pen Input**

The survey also asked students and teachers about their use of the keyboard versus the pen as an input device. Fifty percent of teachers report that they make heavy use of both the keyboard and pen while preparing content for class, while the other 50% report a heavy bias toward the use of the keyboard. However, 80% of teachers report a heavy bias toward the use of the pen during class with an additional 20% of the teachers making extensive use of both the pen and the keyboard during class.

When asked to respond to the question: "To what extent do you use the pen as compared to the keyboard during class?" student responses fall into an almost perfect bell curve. Specifically:

- 10% of students reported exclusive or near exclusive use of the keyboard and 22% reported using both devices but with a heavy bias toward the keyboard,
- 35% reported roughly equal use of both devices,
- 22% reported using both devices but with a heavy bias toward pen, and 11% reported exclusive or near exclusive use of the pen.

#### **4.5 Problems Encountered**

Teachers were asked to indicate their level of agreement with statements about nine problems that might be associated with the use of DyKnow. Teachers rejected eight of the nine potential problems they were asked to consider. Specifically, 10% or fewer of the teachers agreed that these items were problematic. Examples include: overhead of technology, technical glitches, difficulty using the system, and increased time required to use the system. On the other hand, 70% of teachers agreed that student electronic distraction during class was problematic.

Students were asked to indicate their level of agreement with statements about ten potential problems with their use of DyKnow. Thirty-six percent of the students agreed that electronic distraction is a potential problem; and this was the most problematic area identified by students.

Given that the only notable problem identified by DyKnow users relates to electronic distraction it is interesting that only 20% of teachers report using DyKnow's monitoring tools to combat this distraction. Anecdotally we know that DyKnow's monitoring features are heavily used in K-12 environments. In fact, some K-12 teachers see this functionality as the cornerstone of the software solution. The issue of electronic distraction and the related use of monitoring tools is worthy of further study.

## 5. CONCLUSIONS

The survey results clearly illustrate that DePauw's deployment of DyKnow tools has been well received by students with respect to DyKnow's impact on in-class interaction and after-class note review. One student summarized this on the open ended portion of the survey by writing: *"DyKnow really has enhanced my experience here at DePauw. I feel that I was able to study more efficiently due to the notes resulting in higher grades. I also found it very useful to do in class example problems because it gave me a better understanding of the course... I really do not see how I would have gone through out these four years without DyKnow."* A second student reinforced these ideas by writing: *"After taking one CS course without DyKnow, I have definately [sic] come to appreciate it more. I am certain that without it, I would not be able to understand the material in classes nearly as well. I would say the best thing about it is being able to listen and pay attention instead of having to frantically copy down what the professor is saying. It is also helpful to do in class examples and be able to see other peoples' answers. I really feel like DyKnow helps so much in understanding the material."*

The survey results also demonstrate that faculty members place significant value in their use of the DyKnow system. One faculty member explained this on the free response section of the survey by writing: *"I believe my students learn more when I use the system. I also believe class is more enjoyable for both myself and the students when DyKnow Vision is used. I feel more connected to my class, and I believe my students are more connected to me and to the material."*

## 6. REFERENCES

- [1] Anderson R., et. al. Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses, *Proceedings of the 2004 ACM SIGCSE Technical Symposium*, Norfolk, VA, March 3-7, 2004.
- [2] Berque D., Bonebright T., and Whitesell M. Using Pen-based Computers across the Computer Science Curriculum, *Proceedings of the 2004 ACM SIGCSE Technical Symposium*, Norfolk, VA, March 3-7, 2004.
- [3] DyKnow company home page, [www.dyknow.com](http://www.dyknow.com)
- [4] Golub, E. Handwritten Slides on a TabletPC in a Discrete Mathematics Course, *Proceedings of the 2004 ACM SIGCSE Technical Symposium*, Norfolk, VA, March 3-7, 2004.
- [5] Lass R., et. al. *Tools and Techniques for Large Scale Grading using Web-based Commercial Off-The-Shelf Software*, Proceedings of ITiCSE 2003, June 30 – July 2<sup>nd</sup>, Thessaloniki, Greece.

- [6] McFall, R.L., Electronic Textbooks That Transform How Textbooks are Used. *The Electronic Library*, 2005, 23(1): p. 72-81.
- [7] *Transforming Undergraduate Education in Science, Mathematics, Engineering, and Technology*, Committee on Undergraduate Science Education, Center for Science, Mathematics, and Engineering Education, National Research Council, National Academy Press, Washington, 1999.

Published in the Proceedings of CCSCNE-2006/Journal of Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to publish, requires a fee and/or specific permission.