Evolution of a Digital Paper Exam Grading System

Aaron Bloomfield
University of Virginia, aaron@virginia.edu

We describe the evolution of a system designed to allow digital grading, by a human, of traditional paper-based exams or homework assignments. We present a number of new features that build upon the system described in a paper previously published in ITiCSE 2008 [1]. In particular, our system now has the ability to monitor an enormous range of data, from individual question scores, to variances for the same questions across different graders, to question and exam statistics from one semester to the next. The system boasts a number of benefits over paper based grading. Overall grading time is reduced – small courses will see a modest improvement, while large courses can see their grading time cut in half due to the automation of many of the tasks: flipping to the correct page, summing up the pages, recording the grades, returning the exams, etc. Rapid grading of exams can also aid students in a better understanding of the material, as the exam questions are still fresh in their minds when the tests are returned. Simultaneous grading across multiple locations is possible, as there is no single paper copy to hand off to the next grader. Permanent digital records are kept of the exams, allowing for viewing of exams at a later date.

Index Terms – assessment, evaluation, grading, tablet

INTRODUCTION

Educators spend a significant amount of time and energy grading paper-based exams in college courses. Paper exams still remain a primary method to assess student learning in undergraduate curricula, as they have unique advantages that may never be replaced. However, paper exams are time consuming to grade, tally, and record. They are also difficult to keep records of, especially after they are returned to the students. One alternative is electronic exams, which allow many of the time consuming tasks to be automated – entering the grades, returning the exams, etc. However, challenges exist when administrating electronic exams, including ensuring that everybody has a secure computer for the exam. Hence the prevalence of paper exams is expected to continue for quite some time.

The system we present here offers numerous advantages over current grading methods. The time taken to manage the grading (and re-grading) of exams is significantly reduced through the system – even for smaller courses – allowing for more time to be spent by the instructor and teaching assistants on other pedagogically important tasks, such as lecture preparation and course design. Rapid grading of exams can also aid students in a better understanding of the material. Traditionally, paper exams are returned a week (or more) later, causing there to be a temporal disconnect between when a student thought about and answered an exam question, and when they received feedback about the correctness of their answer. With our system, students can receive their graded exam back in much less time – as little 8 (eight!) hours, for example – when the exam question and their response is still fresh in their mind. In addition, the electronic nature of the exams allows for simple archival of the exams, as required by many institutions, especially for final exams. This is particularly relevant as accreditation requirements need to assess student learning on previous exams.

The benefits of this system are not specific to a given discipline. Indeed, our prototype is being used in two other engineering departments. Any paper-based exam can be used with the proposed system. The only requirement is that the bottom 2 inches of each page be unmarked except for the specialized footer.

The system we describe in this paper is a fully functional grading system, to be available soon under the GNU (or similar) license. An exam is created with a specialized footer in the bottom 2 inches of the page, which allows for automated computer recognition. After the students take the exam, it is scanned in via a high-speed scanner, and graded on computers using digital ink via a tablet stylus or a mouse. The grading is done via a web browser, and no specialized software is necessary on the client system; the server system only requires a LAMP (Linux, Apache, MySQL, PHP) architecture. In addition, the system can store the exams locally on the client, to allow for grading when not connected to the Internet. The graded exams are then made available to the students electronically. Many of the time intensive functions that are necessary with paper exams, mentioned above, are no longer necessary.

The merit of this system is to integrate the ease of paper-based exams with the functionality and time saving aspects of electronic exams. The benefits of both systems are merged in a way that advantages both students, faculty, and teaching assistants.

RELATED WORK

There is a growing body of research focused on developing better ways to manage exam grading. This research has focused on a wide variety of topics, from how to interpret exams [2], to automatic grading of programs [3], to detecting plagiarism [4, 5], to grading systems [6]. A

978-1-4244-6262-9/10/$26.00 ©2010 IEEE

similar grading system was presented in [7], but that system focused on grading completely online submissions, not paper-based exams. Indeed, most course management systems have some capability to administer and automatically grade online exams.

There has been much research in tablet-based technology to support pedagogy. Most of this research has been in the context of using tablet computers during lecture [8, 9] or similar areas, such as providing peer-review comments in a CS1 course [10].

Our study adds to the literature by exploring alternative uses of tablets and non-tablets PCs, and by expanding the types of exams that can be graded electronically. We are not aware of any similar systems for grading and management of paper-based exams.

SYSTEM DESCRIPTION

We present a brief overview of the system. For more details, see the full description in [1].

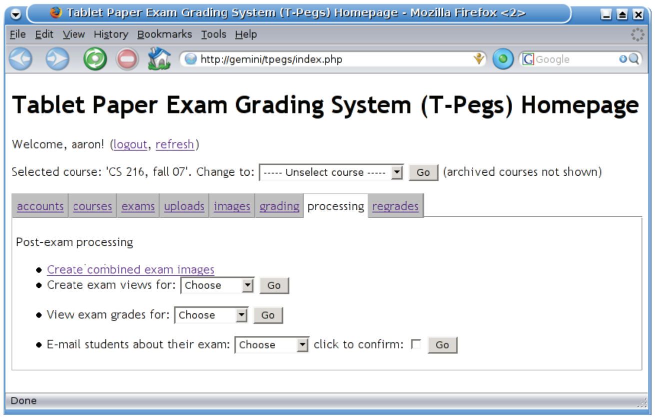
A paper exam is created normally, leaving a 1.75" margin at the bottom. A PDF file of the exam is then uploaded to the system's web-based interface, which adds the footer onto each page of the exam, and returns the resulting exam also as a PDF file. That exam is photocopied normally, and distributed to the students on test day.

This footer can be seen at the bottom of this page, in Figure 1. It contains binary encoded information about the exam on the right side (department, course, exam, page, etc.). On the left side is a space for the students to bubble in their userid. At the University of Virginia, each student has a unique userid that consists of 2 or 3 letters (their initials), followed by a single digit, and then 1 or 2 more letters. The footer shown is thus specific to the University of Virginia, but the ability to have different footers is available, such as if an institution uses 10-digit identification numbers. The students must bubble in their userid on each page of the exam.

The exam is given to the students with the instructions to fill out the bottom bubble footer. Once the exams are collected, the staples are cut off from each exam, and they are then scanned in via a high-speed scanner – modern photocopy machines typically have this capability. A separate high-speed scanner retails for under $1,000; the model we have can scan in 33 double-sided pages per minute. Scanning in 70 10-page exams thus takes about 10 minutes, counting time to reload the scanner input paper tray.

The resulting scans are then uploaded to the system – this can be done via the web-based interface, or through a command-line interface which is a bit faster. At this point, everything becomes digital, and the paper exams are permanently discarded (recycled!).

The course instructor needs to perform some setup for the grading, and this is done through the web interface shown in Figure 2. Indeed, all of the instructor's interaction with the system is through this web interface.


Tablet Paper Exam Grading System (T-Pegs) Homepage - Mozilla Firefox <2>
File Edit View History Bookmarks Tools Help
http://gemini/tpegs/index.php
Google
accounts
courses
exams
uploads
images
grading
processing
regrades
Go
Done

Tablet Paper Exam Grading System (T-Pegs) Homepage

Welcome, aaron! (logout, refresh)

Selected course: 'CS 216, fall 07'. Change to: ----- Unselect course ----- (archived courses not shown)

Post-exam processing

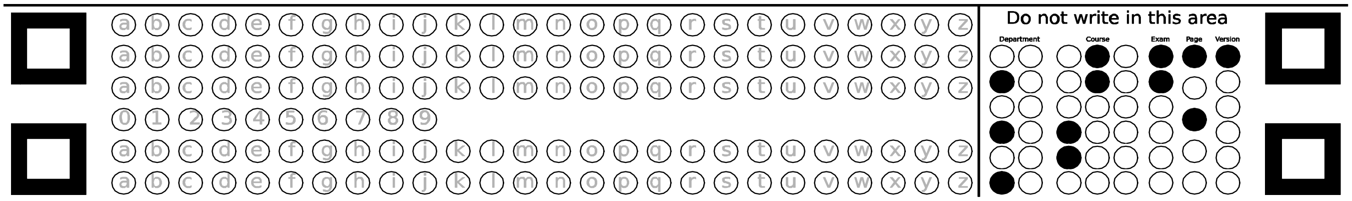
- Create combined exam images
- Create exam views for: Choose
- View exam grades for: Choose
- E-mail students about their exam: Choose click to confirm: □

Figure 2: Web-based instructor interface

In particular, the instructor will direct the system to recognize the bubbled-in footers on the bottom of each exam page. This takes quite some time for the system to perform, but can be left to run unattended. Once completed, any image recognition errors need to be corrected. Such errors can be caused by a student leaving the footer blank, writing too lightly, or entering in the wrong bubbles.

Instructors can set who has the ability to grade the exam. This can be other instructors, graders, or teaching assistants). Once these steps are completed, the exam is ready to be graded.

The system was originally designed to be graded with tablet computers – in portrait orientation – which allow the exam to be seen in almost the same size as the tablet screen. The graders can mark up the exam using the tablet stylus – two of the marks (the second and fourth question in Figure 4) are graded in such a manner. Graders can also enter text (via a keyboard or a tablet's on-screen keyboard), and have the fonts rendered directly on the page – this can be seen in the first and third question in Figure 4.


a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
Do not write in this area
Department
Course
Exam
Page
Version

Recently used textual entries are kept in a drop-down box in the upper right, which is populated dynamically during the grading of a page. This drop-down box is shown in Figure 3. This allows graders to re-enter previous comments without having to re-type the response.

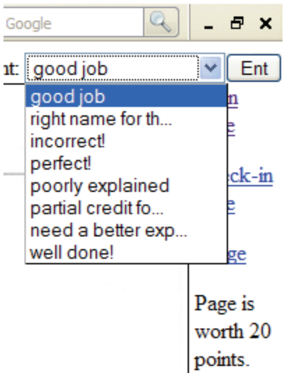
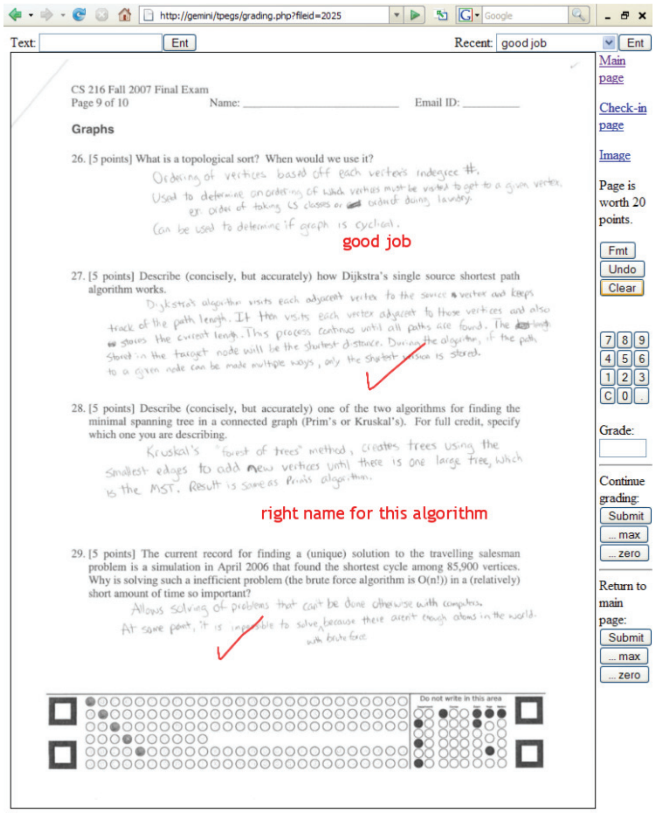


Figure 3: Recently used entry drop-down box

Grades are entered via the keypad on the right side of Figure 4, and once submitted, a new page will appear to be graded. The system supports undoing of markings, as well as clearing of the entire exam page.

The grading interface can be seen in Figure 4, below.



CS 216 Fall 2007 Final Exam
Page 9 of 10 Name: ______________ Email ID: ________

Graphs

26. [5 points] What is a topological sort? When would we use it?

Ordering of vertices based off each vertex's indegree #.
Used to determine an ordering of which vertices must be visited to get to a given vertex.
ex. order of taking CS classes or order of doing laundry.
Can be used to determine if graph is cyclical.

good job

27. [5 points] Describe (concisely, but accurately) how Dijkstra's single source shortest path algorithm works.

Dijkstra's algorithm visits each adjacent vertex to the source vertex and keeps track of the path length. It then visits each vertex adjacent to those vertices and also stores the current length. This process continues until all paths are found. The length stored in the target node will be the shortest distance. During the algorithm, if the path to a given node can be made multiple ways, only the shortest version is stored.

✓

28. [5 points] Describe (concisely, but accurately) one of the two algorithms for finding the minimal spanning tree in a connected graph (Prim's or Kruskal's). For full credit, specify which one you are describing.

Kruskal's "forest of trees" method, creates trees using the smallest edges to add new vertices until there is one large tree, which is the MST. Result is same as Prim's algorithm.

right name for this algorithm

29. [5 points] The current record for finding a (unique) solution to the travelling salesman problem is a simulation in April 2006 that found the shortest cycle among 85,900 vertices. Why is solving such a inefficient problem (the brute force algorithm is O(n!)) in a (relatively) short amount of time so important?

Allows solving of problems that can't be done otherwise with computers.
At some point, it is impossible to solve with brute force because there aren't enough atoms in the world.

✓

Figure 4: Grader interface

Although originally designed for a tablet (and originally designed for that tablet to display the exam in portrait mode), many graders have successfully used the system on a regular desktop PC in landscape mode. Typically, these graders will use keyboard for textual input, and rarely enter line-based mark-ups with the mouse cursor. From anecdotal evidence, speed with a PC is comparable to that with a tablet.

In addition to the variation in class size, the grading varied significantly across the courses. In smaller classes, the instructor would also do the grading. In larger classes, teaching assistants would grade along with the instructor. Some courses also had multiple professors who would pool the grading of exams. Some exams were graded asynchronously, while others had grading sessions of multiple individuals who used the system simultaneously.

Figure 5 shows one of the larger grading sessions that has been held to date using the system. This is actually showing grading being done for two different classes simultaneously, totaling over 250 students. There are 13 graders, both teaching assistants and faculty, that are in the picture, and two more that are out of the camera's field of view. While not all classes are going to have this many students (or this many graders!), it shows that the system can easily scale to a large amount of simultaneous grading. There was no noticeable system delay during this grading session – each page submission takes about 2 seconds of server time, and the pages were submitted asynchronously by the graders.

Figure 5: A very large grading session

The ability to markup an exam page uses the `<canvas>` tag in HTML, which is only present in certain browsers. Currently, only Firefox is used for this grading system, as it has support for this HTML tag. Opera also supports the `<canvas>` tag, but we have not tested the system outside of Firefox. The system works on Firefox on multiple platforms, including Linux, Mac OS X, and Windows.

Once the grading is completed, the instructor has to do a few post-processing tasks. These tasks are similar to the image recognition: they take a while for the system to complete, but can be left unattended.

Finally, the grades are returned to the students. This can be done via an e-mail sent to them with the secure pick-up website location included, or integrated through a course management system (CMS). The latter is the method we use at the University of Virginia, as we use Sakai/Collab for our CMS. The resulting exam is a PDF file that contains the student's scanned in exam (in grayscale) with all the markings overlaid in red, along with a cover page summarizing the score for that exam.

The system also allows for 'regrades' – a comment entered by a student to request a higher grade. These are optional (instructors must actively enable them), and when responded to, will modify both the exam score and the PDF file of the exam that the student receives.

The server runs on a LAMP (Linux, Apache, MySQL, PHP) architecture and is portable to other Unix operating systems – it has also been run successfully on a Mac OS X platform. The majority of the system is written in PHP and provides a web-based interface with all of the system's functionality (see Figure 2, above). The computationally intensive routines, such as the image recognition program, are written in C for execution speed. Javascript is the language used for client-side functionality.

SYSTEM ADVANTAGES

In addition to the ease of grading and the fast turn-around time, there are a number of advantages to this system that are not always readily apparent.

The student's identity can be kept anonymous through the use of a grading mask, which is a gray bar that appears at the top and bottom of each exam page, which is where the student's identifying information is entered. This can be seen below in Figure 6. This gray bar is configurable by the instructor. This prevents the grader from knowing whose exam s/he is grading, thereby eliminating the possibility of bias in the grading. Furthermore, the exams are presented in a random order to the graders. Creation of any grading rubric is often a difficult task, such as with the AP exam. Typically, exam pages graded earlier are graded more or less harshly than exam pages graded later. While this can vary by instructor (some write a complete and comprehensive rubric prior to any grading), this is still an issue in computer science pedagogy. Different graders can have different harshness criteria for grading. The random distribution of pages to a grader allows for increased fairness in the assignment of grades.

Some aspects of cheating (but not all, unfortunately) are completely eliminated by the use of this system. In particular, when a student will leave the answer blank and then fill in the answer afterwards, and claim that their answer was 'missed' during the grading cannot be done with the system, as the student never receives his/her paper exam page back.

In addition, the system can be used with regular homeworks, and is not limited to just exams.

NEW FEATURES

We present two new features not presented before: offline functionality and point-per-question grading.

A useful feature of this system is that an entire stack of exam papers now resides electronically. With the normal mode of using the system, an Internet connection is required for the submitting of an exam page grade and loading of a new exam page.

Offline functionality allows for the grader to download an entire exam (all the student's exam pages) onto a computer, and the grading process can commence without an Internet connection. This allows for grading in situations such as on an airplane, where Internet is generally not available. This functionality uses Google's Gears plug-in to Firefox [11]. Google is stopping support for Gears in the near future, as the ability to load files onto a computer will be replaced as a feature in HTML5. Once HTML5 is standardized in browsers, the system will be updated to use that ability as well.

Since there is no Internet connection, some concurrent features of the system – such as preventing two graders from grading the same page – cannot be performed. Generally, the offline feature is used either when a grader is grading the entire exam himself/herself, or when multiple graders have previously split up who is grading which pages on a given exam.

System response in offline mode is actually faster than online mode, as the computer does not have to wait for the data to be transmitted over the network. Once re-connected to the network, the data is easily uploaded to the server.

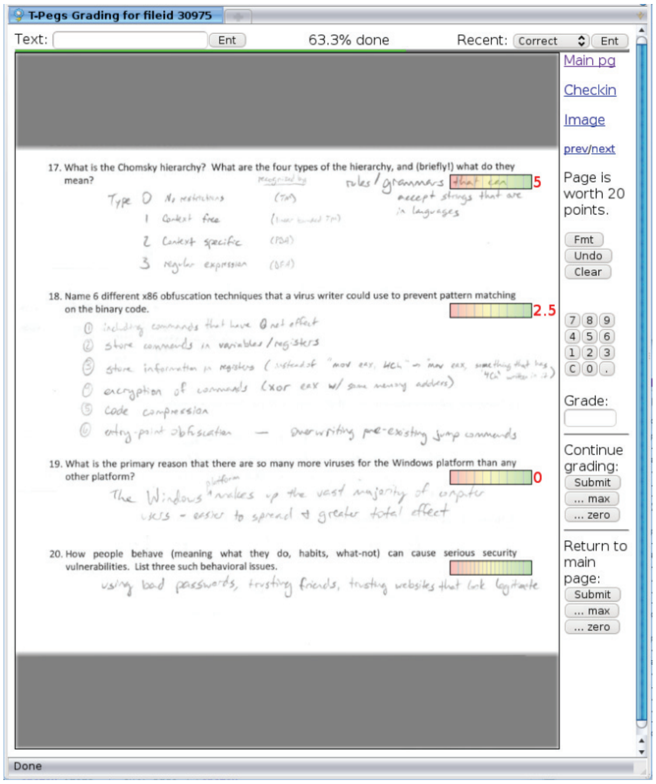
The second new feature is per-question grading. Previously, exams were graded on a per-page basis: meaning a score was assigned to each page, and not to individual questions. The current system has the ability to award points on a per-question basis. This can be seen below in Figure 6.

A grayscale reproduction of this article may not render that image correctly, but the four boxes on the right side of the exam page are a gradient that transition from red on the left, through yellow, to green on the right. The numbers to the right of the top three of these boxes are in red, and are entered by the grading system.

With this functionality, each grader will click on the 'point box' to award a value for that individual question; the points awarded for each question can be changed simply by tapping the box again. Tapping towards the right will award more points, while tapping towards the left will award fewer points. Once all the points have been awarded, the grade is automatically filled in on the form on the right, and the page grade can then be submitted.

This feature allows for faster grade entry, as it only requires a single click instead of writing the points awarded for each question. Furthermore, the points awarded on each question are saved, and can be analyzed later. This allows for examining which questions students did the best or worst on, as well as comparisons between multiple graders on a single page.

Unfortunately, this feature is so new that we do not yet have data to present statistics on it.



17. What is the Chomsky hierarchy? What are the four types of the hierarchy, and (briefly!) what do they mean?

rules / grammars that can accept strings that are in languages

recognized by

Type 0 No restrictions (TM)
1 Context free (linear bounded TM)
2 Context specific (PDA)
3 regular expression (DFA)

18. Name 6 different x86 obfuscation techniques that a virus writer could use to prevent pattern matching on the binary code.

① including commands that have 0 net effect
② store commands in variables / registers
③ store information in registers (instead of "mov eax, 4Ch" → "mov eax, something that has '4Ch' written in it")
④ encryption of commands (xor eax w/ some memory address)
⑤ code compression
⑥ entry-point obfuscation — overwriting pre-existing jump commands

19. What is the primary reason that there are so many more viruses for the Windows platform than any other platform?

The Windows platform makes up the vast majority of computer users - easier to spread + greater total effect

20. How people behave (meaning what they do, habits, what-not) can cause serious security vulnerabilities. List three such behavioral issues.

using bad passwords, trusting friends, trusting websites that look legitimate

Figure 6: Grading page with masks and point-per-question boxes

DATA COLLECTION

The system collects a significant amount of data. In addition to keeping permanent electronic copies of each exam, the score for each individual page is recorded, along with who graded that particular page.

With the introduction of the point-per-question functionality, data is now kept for each question graded on each exam. This allows the instructor to see exactly on which questions the students performed well, and on which questions the students performed poorly. Furthermore, variances across graders for the same page and questions can easily be obtained, so as to compare the grading of different graders. This is especially useful in very large classes, where a given page is graded by multiple graders.

If questions across semesters of the same course are similar enough that they can be directly compared (perhaps the same question, or a different question but on the same topic and with the same level of difficulty, etc.), then this data can be used to directly compare students between multiple semesters of the same course.

All the exam scans, as well as the digital mark-ups, are stored electronically. While the storage requirements are not small (average of 1 Mb per exam page), it is certainly within modern reasonable server storage requirements. The storage of the exams allows for easy record keeping, even after the exam has been returned to the students.

Accreditation requirements often require departments to look at previous exams to evaluate student performance, student learning, and other aspects. Use of this system allows for this to be done easily, as both the exams and the points awarded per question are kept long after the students have finished the course.

In an effort to preserve student privacy, and to in accordance with FERPA, student records are only available to the student and the instructor for that class.

RESULTS

As of the spring of 2010, the system has been used in 37 different courses with 110 exams (and other assignments). Course sizes ranged from 11 students to 501 students, the latter in a CS 1 class. The average class size was 107 students, and it was 81 if the two 500-student CS1 classes are eliminated. About 4,000 students have taken tests using the system, but 2,700 unique users (students may take exams in this system across as many as 7 different classes). There are 9 faculty members actively using the system; most are in the Computer Science Department, but two other departments in the Engineering school are represented as well: Mechanical Engineering and Engineering Science. There are over 70,000 exam pages entered into the system.

The actual act of grading – as opposed to the other administrative tasks – is generally as fast as, if not faster than, grading a paper-based exam. This was measured subjectively – all participants who were asked this question, both teaching assistants and faculty, agreed on this claim. While the time to read a page and determine a grade is on par with the time taken on a paper-based exam, the act of moving to the next page is expedited significantly – it takes about 2 second to load up the next page electronically, versus the time spent moving to the next page with a paper exam.

The administrative tasks associated with administering an exam are dramatically reduced. There certainly is overhead with this system – scanning in the exams and fixing the image recognition errors, in particular. However, this is offset by the fact that most of the tasks that are necessary for paper exams are completely handled by the system, including: tallying up each page, summing those results (and then checking that sum), entering it by hand into a spreadsheet, and returning the exam.

The turn-around time – from when an exam is completed until it is returned to the students – can be dramatically reduced. In the fall of 2009 in a class of 64 students, the final exam was returned less than 8 hours after the students finished the exam. While there were multiple people helping grade the exam, it was still a dramatic turn-around time for an exam.

Different instructors also had different methods for grading: some used tablet computers, while others used their desktop or notebook computers. For those that used tablets, the system used were 1 GHz Pentium machines, with 512 Mb RAM, 768x1024 resolution, and Windows XP – these are the tablets shown in Figure 5. These tablets are slow by current standards, which caused the grading to take longer than it would have otherwise. The server was a 3 GHz Pentium computer with 2 Gb RAM running Debian Linux. The tablets were connected to the server via a wired network and a dedicated Ethernet switch – we used a wired network for speed and stability reasons.

We found that image recognition errors – errors recognizing the bubbled userids on the exam footers – ranged from 0% to 10% of the scanned exams, but averaged about 7%. The variability depended largely on how well the students were instructed to enter their userids.

The first use of the system took some additional time, for both faculty and the graders, to learn the system interface and become accustomed with entering the marks on the exams. This was not unexpected, as any new system has a learning curve. Successive grading sessions eliminated this training time.

Subjectively, the graders and faculty involved felt the system made the entire process easier to manage and grade the exams. Indeed, faculty with classes as small as 11 students used the system. Other instructors who used the system reported a perceived decrease in overall grading time ranging from a *minimum* of 25% up to 50%. This was true for both large courses that had multiple graders and individual instructors who were grading the exam alone. The decrease in time was due to the fact that many of the tasks required for paper-based exams were done automatically (summing up the grades, etc.).

Objectively, there is little additional data to provide, other than the image recognition error rate described above. This system is designed to make the grading process easier, and the success of this goal is provided mainly through subjective assessment.

FUTURE WORK

This is an evolving system, and there are many improvements that we want to make.

We plan to shortly release the source code into the public domain under a GNU (or similar) license. We have some issues that we plan to complete prior to this: a few stability fixes, better documentation, a proper unit testing framework, and commenting of the code.

Much of the functionality of the system requires specific software – Firefox for the browser, and Google's Gears plug-in for the offline functionality. While Firefox is readily available (and easy to install!), we plan to make this system work on multiple browsers (Opera, Internet Explorer, etc.). With the adoption of HTML5, the code that uses the Gears plug-in can use the new features of HTML5 to implement the same offline functionality.

Increased adoption is a focus as well. We plan to scale up usage of our system: first growth within our department, then within our school, then to other institutions. As the system grows, an online community will grow to support it as well.

REFERENCES

[1] A. Bloomfield and J. F. Groves. "A tablet-based paper exam grading system." In *ITICSE '08: Proceedings of the 13th annual SIGCSE conference on Innovation and technology in computer science education*, New York, NY, USA, 2008. ACM.

[2] P. Thomas, N. Smith, and K. Waugh. "Computer assisted assessment of diagrams." *In ITICSE '07: Proceedings of the 12th SIGCSE conference on Innovation and technology in computer science education*, pages 68–72, New York, NY, USA, 2007.

[3] M. T. Helmick. "Interface-based programming assignments and automatic grading of java programs." *In ITiCSE '07: Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 63–67, New York, NY, USA, 2007. ACM.

[4] J.-H. Ji, G. Woo, and H.-G. Cho. "A source code linearization technique for detecting plagiarized programs." *In ITiCSE '07: Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 73–77, New York, NY, USA, 2007. ACM.

[5] S. Schleimer, D. S. Wilkerson, and A. Aiken. "Winnowing: local algorithms for document fingerprinting." *In SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85, New York, NY, USA, 2003. ACM.

[6] R. E. Noonan. "The back end of a grading system." *In SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 56–60, New York, NY, USA, 2006. ACM.

[7] J. L. Popyack, N. Herrmann, B. Char, P. Zoski, C. Cera1, and R. Lass. "Pen-based electronic grading of online student submissions." *Syllabus*, 18-20, January 2003.

[8] R. Anderson, R. Anderson, K. M. Davis, N. Linnell, C. Prince, and V. Razmov. "Supporting active learning and example based instruction with classroom technology." *SIGCSE Bulliten*, 39(1):69–73, 2007.

[9] D. Berque, T. Bonebright, and M. Whitesell. "Using pen-based computers across the computer science curriculum." *SIGCSE Bulliten*, 36(1):61–65, 2004.

[10] E. F. Gehringer. "Electronic peer review and peer grading in computer-science courses." *Proceedings of the 32nd SIGCSE technical symposium on Computer Science Education*, pages 139–143, February 2001.

[11] Google gears plug-in for firefox. Google, 2009. http://gears.google.com/.