

COMBINING, STORING, AND SHARING DIGITAL INK

Jesse M. Heines and Lucy W. Liang
Dept. of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
978 934-3634
{heines, wliang}@cs.uml.edu

ABSTRACT

A number of applications are available that are designed to enhance student engagement in the classroom through the use of Tablet PCs. The value of this goal is well documented, and researchers working with applications such as Classroom Presenter and DyKnow Vision have blazed the way in studying not only how best to *deploy* this technology, but also how to best *teach* with it in actual classrooms. Our own experiments with Tablet PCs using Classroom Presenter have identified three issues that we address in this paper: (1) the combining of digital ink from multiple students into a single display so that one can quickly get a sense of the overall classroom response, (2) the storing of digital ink so that it can be recalled later in a structured manner, and (3) the reliable sharing of digital ink between systems so that absolutely no strokes are lost due to transmission errors.

ENGAGEMENT VIA DIGITAL INK

Professors often try to engage students by asking questions. It is not uncommon, however, for only a few students to respond. Some remain silent because they're afraid to make a mistake in front of their peers. Some hate drawing attention to themselves. Some lack confidence in their English. And some, especially those brought up outside the U.S., come from cultures that eschew speaking unless spoken to directly, particularly when in the presence of an authority figure such as a professor.

Unfortunately, students who do not engage in class discussions often "tune out" and withdraw from the underlying thought processes as well, not even thinking about the questions being asked or trying to answer them for themselves. Instead, they simply wait for another student to respond. Such behavior causes students to miss out on the reasoning, concluding, erring, and analyzing of one's mistakes that are essential parts of the learning process.

It is widely recognized that effective questioning strategies enhance learning, not only by getting students to think, but also by getting them engaged in the larger classroom process. (See, for example, Cotton's comprehensive study of classroom questioning [6].) Even in predominantly lecture-based classes, Hughes [8] points out that "the lecture format ... was never intended to be a boring 'telling' to a bored 'listening' audience."

Numerous professors have experimented with having students respond to questions through Tablet PCs [7,9,10] using such tools as Classroom Presenter [1] and DyKnow Vision [4]. These tools allow instructors to view individual student responses to questions in the form of ink markup on a slide generated by PowerPoint or a similar tool [10]. Two major advantages of this approach are anonymity, which removes fear, and increased richness over simple verbal responses through the ability to draw and fill in instructor-supplied templates.

We have done likewise, attempting to use Classroom Presenter with up to 27 Tablet PCs. Our experience has revealed three important issues.

(1) Classroom Presenter's "slide deck" metaphor for student submissions, in which each appears to the instructor as an individual slide that can be displayed to the entire class for discussion, quickly becomes unwieldy as class size increases. This paradigm makes it difficult to get a feel for how well the class *as a whole* understands the topic being taught, leaving us guessing whether to go on or to spend time reviewing.

(2) There is no way to store student responses in a structured manner. One can store JPG images of screens with digital ink, but this loses the ability to analyze or manipulate student responses in any way.

(3) Ink loss between Tablet PCs is common and worsens dramatically as the number of PCs increases into the teens. We found this issue to be particularly disruptive to the class, as students got "hung up" when they saw only partial ink. They continually checked with each other to see if their screens were the same, and when they were not they'd call out that they weren't seeing all the ink, making it hard to proceed.

The remainder of this paper describes the approaches we have explored to try to address these Classroom Presenter issues. This work was done in the context of a research project. DyKnow Vision has also addressed them in the context of a proprietary commercial software system.

COMBINING DIGITAL INK FROM MULTIPLE STUDENTS

Figure 1 demonstrates Classroom Presenter's slide metaphor for submitted student responses. We found this to be a highly valuable feature because it allowed us to zero in on the specific misunderstandings that one or more students sitting in that particular class have at that very point in the lecture, explaining for the benefit of all why certain ones are incorrect. However, we also felt handcuffed by this feature because the slide deck metaphor only allowed us to view one student response at a time.

Consider, for example, the simple problem shown in Figure 1. To get a feel for just how many or what percentage of responses are incorrect, the instructor must scroll through each response in turn, a time-consuming process from which one still may not be able to see any pattern in student responses. To address this issue, we wanted to implement the ability to overlay ink from multiple students as shown in the *mockup* in Figure 2. (N.B. Figure 2 is *not* a capture of an actual Classroom Presenter interaction!)

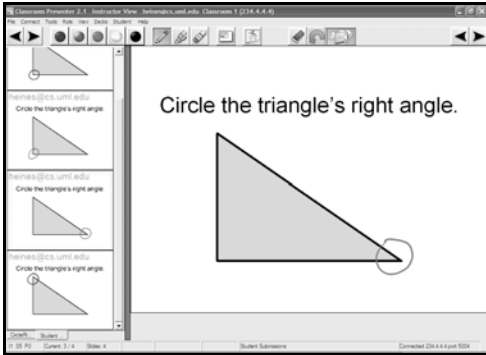


Figure 1. Instructor view of an (incorrect) individual student response in Classroom Presenter.

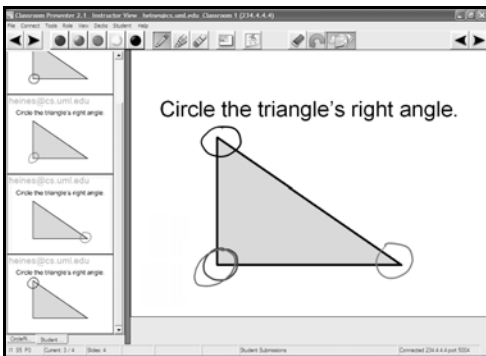


Figure 2. *Mockup* instructor view of four overlaid student responses in Classroom Presenter.

We developed an approach to overlaying student responses by serializing the digital ink they created, transmitting it to a web server, storing it on the server as byte strings, and then concatenating any combination of those strings on request and returning the entire construct to client systems as a single set of ink strokes. (The technical details of this approach are discussed in the next section.)

The results we were able to achieve are demonstrated in Figures 3 and 4, which are *actual* screen captures from our system (called the *Engaged Classroom*). Figure 3 shows the student view of a single response, using course content provided by Jamie Rogers of Murray State University (Kentucky) for experimentation with our system. This is a correct response to an exercise in a statics course (the study of physical forces), where the student has calculated the magnitude of the force as the area under the triangle, *i.e.*, $\frac{1}{2} \times 3 \text{ ft} \times 2000 \text{ lb/ft} = 3000 \text{ lb}$, and has indicated that the triangle's centroid is located $\frac{2}{3}$ of the distance from the base from its end.

Figure 4 shows the instructor view when responses from multiple students are overlaid. While it is obvious that overlaying student-entered text can easily become gibberish, the experienced eye of an instructor can quickly pick out trends in the overlaid student-entered graphics. If the instructor wishes to discuss a specific response with the entire class, that response can be displayed on its own by checking only the corresponding checkbox in the Connected Sessions list at the upper right of the window.

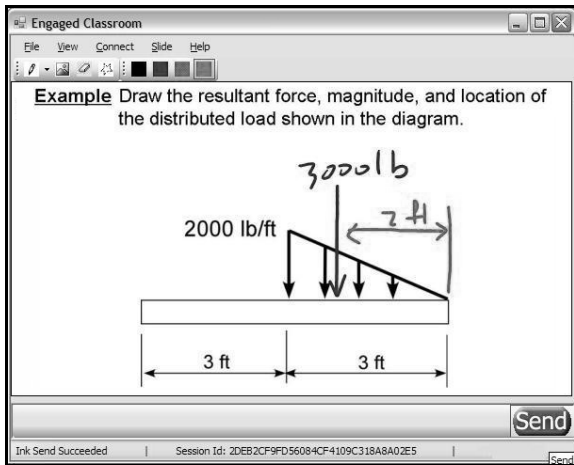


Figure 3. Student view of a (correct) individual student response in the Engaged Classroom.

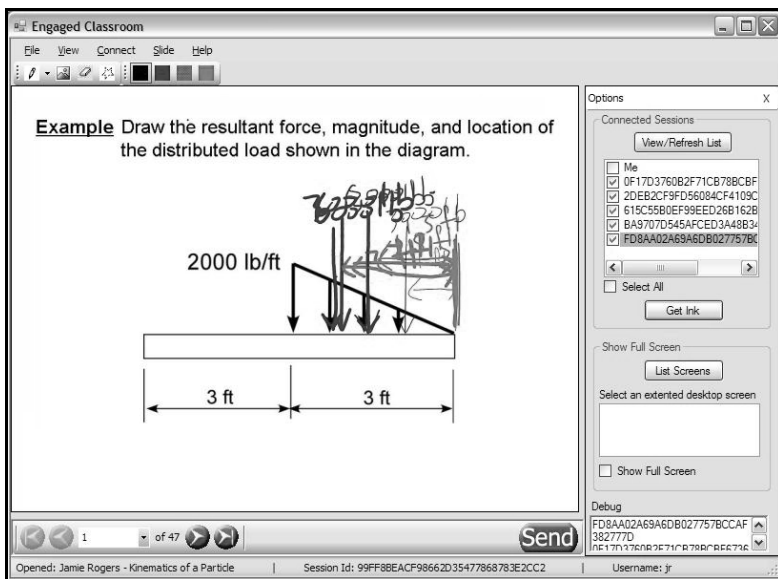


Figure 4. *Actual* instructor view of four overlaid student responses in the Engaged Classroom.

STORING DIGITAL INK

One of the most important consequences of working with digital ink in serialized form is the ability to store it in a standard database and then access it via SQL. This allowed us to combine responses as described in the previous section and would also allow highly efficient retrieval of students' responses to "replay" and study classroom interactions after the fact.

The schema we use for the `inkdata` table in our MySQL database is shown in Figure 5. Each record stored one *part* of a student's ink response, and most fields should be self-explanatory. The one really tricky problem we had to deal with is that we got unreliable results when we transmitted greater than 4096 bytes to the server.

Name	Type	Length	Decimals	Allow Null
RecordNo	int	10	0	<input type="checkbox"/>
CreationDateTime	datetime	0	0	<input type="checkbox"/>
LongDate	bigint	20	0	<input checked="" type="checkbox"/>
SessionID	varchar	255	0	<input type="checkbox"/>
Cleared	enum	0	0	<input type="checkbox"/>
QuestionNo	int	11	0	<input type="checkbox"/>
PartNo	int	11	0	<input type="checkbox"/>
CourseID	varchar	255	0	<input checked="" type="checkbox"/>
ClassID	varchar	255	0	<input checked="" type="checkbox"/>
SlideFileName	varchar	255	0	<input checked="" type="checkbox"/>
InkData	text	0	0	<input checked="" type="checkbox"/>

Figure 5. Schema for the `inkdata` table in the Engaged Classroom MySQL database.

We addressed this by breaking long serializations into “chunks” that did not exceed this limit. The PartNo field helped keep everything straight so that each student’s complete response could be reconstructed through concatenation.

In addition to ease of retrieval, storing digital ink in serialized form in a standard database is highly efficient and allows the data to be backed up quickly and easily with standard dump and load routines. We have found the performance and convenience of the MySQL database to be highly satisfactory for our purposes.

There are, of course, other tables in the database to support multiple courses and instructors who must log in to set up classes and perform tasks such as uploading class slides to the server. (Unlike Classroom Presenter, the slides we use are standard JPG files.) We have also developed a web interface to simplify instructors’ access to the database.

RELIABLE SHARING OF DIGITAL INK

One of the FAQs posted on the Classroom Presenter website states: “Presenter is built on top of multicast communication, which does not guarantee delivery. This means that sometimes the information displayed on the viewer will differ from the information displayed on the presenter. The two types of loss you might observe are slide loss, where some of the broadcast slides don’t get through, and ink loss, where occasional ink strokes are dropped” [2]. Ink loss distracted students so severely that it could stop the class cold. It was particularly frustrating that the degree of ink loss seemed to vary randomly throughout the class, both in terms of what was missing and on which system it was missing.

In addition to Anderson, Classroom Presenter’s developer at the University of Washington, experiencing these problems himself, others have reported them in postings to the “presenter-profs” mailing list (<https://mailman.cs.washington.edu/mailman/listinfo/presenter-profs>) and informal discussions at conferences. Despite help from Anderson and developers in Microsoft’s ConferenceXP Group, we were unable to resolve these issues and get more than about ten systems working together reliably. This is what drove us to look for a more reliable way to transmit digital ink between systems. It is interesting to note that Anderson is also addressing these issues by introducing TCP/IP-based client server communication (in addition to multicasting) in Classroom Presenter 3 [3].

As discussed earlier, the key to our approach is serialization. When students draw on their tablets and click the Send button, their ink is serialized and transmitted to the server along with various identifiers using standard HTTP protocol. This byte stream is read and parsed by the server and stored as text in the MySQL database record discussed in the previous section.

A detailed discussion of the internals of our programs is beyond the scope of this paper, but two key points are noteworthy.

(1) The server-side program is a JavaServer Page (JSP) that uses standard request.getParameter calls to parse incoming data. All processing is done within a single JSP to increase the probability that this page will remain in server memory, thereby achieving maximum performance. Connection pooling is used to maximize database access efficiency. The server-side program recognizes a number of commands to perform various tasks at the request of the client.

(2) The client-side program is written in C#. It serializes ink entered by the user to convert it to an encoded string that is passed as an argument to the WebClient.DownloadData method for transmission to the server. This method is also used to retrieve responses from the server in the form of a byte array, which is then interpreted by the client. Ink is transmitted to the server in “chunks” no larger than 4096 bytes so as not to surpass the server’s maximum URL length. When the client subsequently requests ink to be transmitted from the server, the chunks are reassembled into a single byte stream before transmission, thereby making it easy for the client to reconstruct a perfect copy of the original ink.

Using serialization, ink transmitted by a client, stored on the server, and transmitted to other clients (on request, see below) retains all of its native ink qualities, including thickness variations caused by varying stylus pressure. We have been able to achieve perfect reproductions of ink transmissions totaling tens of thousands of bytes in serialized form, and we know of no limit to the number of strokes, complexity of the drawing, or size of the serialized ink string that can be reproduced perfectly.

COMPARING APPROACHES

Our approach is similar to that of Wilkerson *et al.* [12] in the Ubiquitous Presenter (UP) system developed by Beth Simon and William Griswold at the University of California, San Diego. Both employ a client program running on Tablet PCs to transmit ink to a web server and then distribute that ink to other PCs from the server. The major differences in our approaches are as follows.

- (1) UP uses a modified version of Classroom Presenter as its client, while we wrote our own custom client.
- (2) UP reproduction of student-entered ink results in graphic lines of the same quality as those drawn with a mouse, while we reproduce actual digital ink. (This UP limitation is due to the fact that it supports non-tablet devices through a Java applet.)
- (3) UP stores ink reproductions in JPG files, while we store ink in a MySQL database.
- (4) UP can work with standard laptops, while we require students to use Tablet PCs.

(5) UP is a proven system, has functionality beyond the scope of this paper, and has been used successfully in actual classrooms, while our work is still in the experimental stage. (Simon [11] reports that UP will support 30-40 active classrooms in the Fall 2006 semester.)

It is important to note that because our approach, like that of UP, is web-based, digital ink cannot be “broadcast” to clients from the server as it is by Classroom Presenter, which uses multicasting. In our approach, the client must actively request an update from the server. We therefore implemented a polling feature whereby clients automatically request updates every 3-5 seconds. Given the built-in capabilities of web servers to handle multiple clients simultaneously, we have found that responses to client requests result in reliable digital ink transmission even when the server is running on a laptop system in the classroom.

The trade-off, then, seems to be between instantaneous ink transmission and reliable ink reproduction. If the former is a critical factor, the multicast approach will be required. But if the latter is more important and a few seconds’ delay can be tolerated, the polling approach may be more desirable.

FUTURE DIRECTIONS

As noted in the previous section, our work is still in the experimental stage. As of this writing in September 2006, our tools are not yet ready to be released for use by others. Further testing of our approach is needed, and there are several improvements that we would like to make in the various user interfaces employed by the Engaged Classroom’s components. Thus, our work should not be construed as an effort to compete with Classroom Presenter or any other existing system, but rather as an investigation of techniques that might address certain problems inherent in the use of some of those systems.

That said, we feel that this work demonstrates a viable and important approach to the sharing, combining, and storing of digital ink, and one that could greatly reduce the frustration caused by the low reliability of multicast communication employed by some existing systems. The storage and transmission techniques described in this paper are 100% reliable with large numbers of systems while still maintaining the ability to work with real digital ink, complete with all of its subtle characteristics.

Rather than developing yet another system, we would like to see this work incorporated into the capabilities of existing systems. Although the Classroom Presenter source code is available for download from its website, it has not proved easy to work with. We therefore developed our own code to experiment with the techniques discussed in this paper. While we have been unable to collaborate with other developers and researchers at the source code level to date, we hope to do so in the future and make our source code similarly available for to others.

We have not worked with DyKnow Vision due to its proprietary status, but it is interesting to note that that system uses a similar approach. Berque [5] reports that DyKnow Vision “...uses HTTP as its communication protocol. The server uses a SQL server database and data is broken into appropriately sized chunks. Because of this, DyKnow does not drop pages or ink strokes at any time. The system is routinely used in classes with 30+ students and has been used in classes with 100+ students as well.”

The database approach has additional promise beyond the storing of digital ink. In a previous project, we implemented the ability for students to respond to class polls and open-ended questions. Figure 6 shows how responses to a simple Yes/No question appear to an instructor and would be projected for the class to see. Figure 7 shows how responses to a more complex question appear. It can be seen that even with 18 responses, it is quite easy to scan these quickly to get a feel for overall class comprehension and to zero in on specific incorrect responses. We hope to integrate these capabilities into the current version of the Engaged Classroom as well.

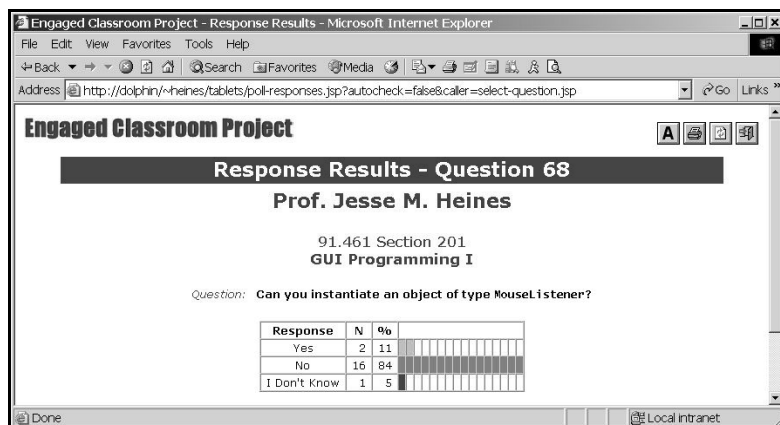


Figure 6. Response results for a simple Yes/No question in the Engaged Classroom.

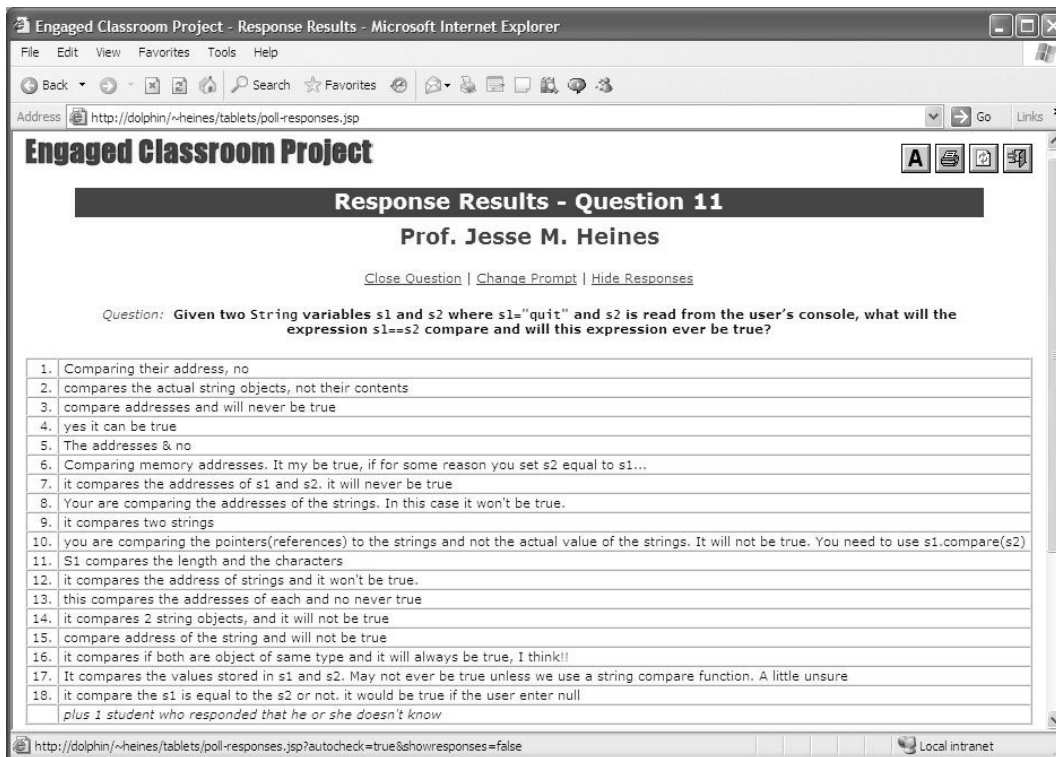


Figure 7. Response results for a freeform question in the Engaged Classroom.

ACKNOWLEDGMENTS

This work was supported in part by a University of Massachusetts Strategic Initiative Grant. We gratefully acknowledge the contributions of Jamie Rogers of Murray State University (KY) for providing us with sample course materials with which to test our system. We also acknowledge the contributions of Richard Anderson of the Univ. of Washington, Beth Simon of UCSD, Dave Berque of DePauw Univ., and especially Fred Martin of UMass Lowell for reviewing drafts of this paper and making important suggestions for improvement.

REFERENCES

- [1] Anderson, R. Beyond PowerPoint: Building a new classroom presenter. *Syllabus*, pp. 31-33, June 2004.
- [2] Anderson, R. Classroom Presenter FAQs. www.cs.washington.edu/research/edtech/presenter/faq.html, accessed August 21, 2006.
- [3] Anderson, R. Personal correspondence. September 1, 2006.
- [4] Berque, D. An evaluation of a broad deployment of DyKnow software to support note taking and interaction using pen-based computers. *Journal of Computing Sciences in Colleges* 21(6):204-216, June 2006.
- [5] Berque, D. Personal correspondence. September 5, 2006.
- [6] Cotton, K. Classroom questioning. *School Improvement Research Series*, Northwest Regional Educational Laboratory, 2001. www.nwrel.org/scpd/sirs/3/cu5.html, accessed Aug. 21, 2006.
- [7] Cox, R. and Rogers, J. Enter: the (well-designed) lecture. *The Teaching Professor* 19(5):1,6, May 2005.
- [8] Hughes, J.A. Is lecture format still an effective teaching tool? *NEA Higher Education Advocate* 22(1):11, 2004.
- [9] Razmov, V. and Anderson, R. Pedagogical techniques supported by the use of student devices in teaching software engineering. *Proceedings of the 37th Technical Symposium on Computer Science Education*, Houston, TX, pp. 344-348, 2006.
- [10] Simon, B., Anderson, R., Hoyer, C. and Su, J. Preliminary experiences with a Tablet PC based system to support active learning an computer science courses. *Proceedings of the 9th Annual Conference on Innovation and Technology In Computer Science Education*, Leeds, U.K., pp. 213-217, 2004.
- [11] Simon, B. Personal correspondence. September 3, 2006.
- [12] Wilkerson, M., Griswold, W., and Simon, B. Ubiquitous Presenter: increasing student access and control in a digital lecturing environment. *Proceedings of the 36th Technical Symposium on Computer Science Education*, St. Louis, MO, pp. 116-120, 2005.