

# Captured Presentations for Online Learning

**Curt Hill**

Department of Mathematics  
Valley City State University  
Valley City, ND 58072  
[Curt.Hill@vcsu.edu](mailto:Curt.Hill@vcsu.edu)

**Brian M. Slator**

Department of Computer Science  
North Dakota State University  
Fargo, ND 58102  
[slator@cs.ndsu.edu](mailto:slator@cs.ndsu.edu)

## Abstract

Students like and expect the use of presentation software, such as Microsoft's PowerPoint®. The importance of this type of software cannot be underestimated, nor can the controversy concerning its effectiveness. This paper provides a brief review of the controversy and considers the integration of presentation software into an online learning system as well as its use in a classroom course. This involves a Java applet that displays the slides and plays the accompanying lecture. The student may then view the slides while hearing the lecture either from within an enhanced hypertextbook or a web page. The details of this implementation are discussed.

## **Introduction**

Presentation software pervades both the business and government sectors. Is this a technological fad or a significant advance in communication? Neither viewpoint is lacking for supporters, although it seems likely that the truth is somewhere in between. When used properly such software adds to the presentation and when used poorly it obscures information and hinders the understanding of the audience members. The next section provides a brief review of the literature concerning the effectiveness of software of this type.

In order to capitalize on the widespread use of PowerPoint the Valley City State University MOO Educational Platform system (formerly known as the ProgrammingLand MOOseum [7,8,9]) was modified to support the integration of audio and graphics from a presentation. The VCSU MEP system focuses on tracking student progress through a set of lessons. The presentation voice-overs may be made a lesson requirement, thus the system itself verifies that an individual student has reviewed the entire presentation. This system is reviewed in the third section and the implementation of the presentation viewer is covered in the fourth section. The fifth section considers the restriction when the recorded presentation is outside of the MOO environment.

## **Presentation Software**

In a little over a decade, presentation software has revolutionized the time honored lecture. PowerPoint made its appearance in 1987 but previously had been named Presenter[1, 13]. It was not until it became part of the Microsoft Office Suite™ that ubiquity started to occur. Although other products exist, they now possess such a small market share that the word PowerPoint has become a genericized trademark (like aspirin) and has come to refer to either a presentation or the software that displays it.

Like most revolutions there have been both detractors and proponents. Perhaps foremost among the critics is Edward Tufte[15]. He is not so much against this type of software, but against how its predefined formats have dominated how presenters have used it. “PowerPoint is a competent slide manager and projector. But rather than supplementing a presentation, it has become a substitute for it.”[16] Dale Cyphert echoes this: “Requiring students to create and use electronically generated “visual aids” seems to result in a semester-long demonstration of things speakers should never do with projection equipment.” [5]

This is not necessarily the case for an experienced classroom teacher. Although the creation of the initial presentation requires some effort [10], it has some advantages. Experienced lecturers who are quite familiar with the content may overlook reminders in

their notes, while displaying a slide generally keeps them on task. If the teacher is familiar with the material the bulleted points are all that is needed to remind them of the things that need to be discussed. Graphics – such as photographs or diagrams – may be displayed easily and without the physical management needed for transparencies. A question or missed point often prompts the instructor to an upgrade of the presentation. Presentations tend to improve over time.

Thus, while Tufte is widely considered an expert in the field of presentation communication[17], the use of this software is widespread. The obvious fact is that it is generally much easier to create and deliver a presentation with PowerPoint than using a lecture accompanied by overhead transparency slides or markings on a chalk board.

Meanwhile, studies on computer generated slides get mixed reviews. (A good and recent review of this literature is in Lavasseur and Sawyer[11].) There are several arguments that suggest why a computer presentation is valuable. Generally, the students appreciate that the professor is using computer-generated slides[3]. The slide and lecture format appeals to multiple learning styles[3].

There are several negatives as well. Students often prefer the current educational fad[11], whether it increases learning or not. Students may even prefer an approach that measurably diminishes learning[11]. It is also possible, even easy, to overload the visuals in a way that inhibits learning. The students become so distracted by the visuals they miss the important content. There is a tendency for a presentation to make the presenter more distant[11] as well as to suppress discussion[6].

Every medium has strengths and weaknesses. Microsoft introduces new features into PowerPoint in order to increase profit, rather than to make it more effective in an educational or business environment. It is the responsibility of the classroom instructor to use their tools effectively. The usual expert suggestions include:

- Use a high contrast between text and background so that the screen is easy to read.
- Avoid the bells and whistles that detract from the content. Otherwise the student devotes too much attention to details which are of no value.
- Avoid too much text and small fonts, because difficulty in reading the screen detracts from learning.
- Make the presentation available on the web or provide handouts. The students will then have an easier time focusing on the content rather than trying to capture the outline shown on the screen.

On this final point, Lavasseur and Sawyer[11] observe that several studies which show increased student learning and good experimental design had the last point in common as well. It would seem likely that giving the students handouts or allowing them to print the

slides from the web is beneficial. Apparently this form of scaffolding for recording notes is advantageous, either by itself or in combination with the presentation.

## **MOO Educational Platform**

The VCSU MEP (formerly known as ProgrammingLand) [8] is an active learning hypertextbook[2] providing content material for Computer Science courses.

VCSU MEP has replaced a textbook in regular classroom courses and as the principle resource in online courses. The underlying technology is that of a MOO [4], which is an object oriented MUD. MUD stands for Multiple User Domain which includes most of the multiple-user, online, text-based games. The system contains content comparable with a textbook, yet also provides many features of a Learning Management System such as online tests, quizzes and surveys. VCSU MEP content is organized in terms of lessons, where each lesson contains a list of activities that a student must complete and which the system records. Certain lessons have an external assignment, which is often a programming assignment but may be the completion of a worksheet or production of an essay. When a student completes the lesson, an agent delivers the assignment to the student. Unlike using a textbook, the student who does not perform the reading and other activities receives no assignments. The system monitors the progress of the students as well and may even dispatch an agent to help students to get back on track [9]. Like any MOO, students are aware of each other and may chat anywhere they may meet.

The client for this system is any Java-enabled web browser, which allows the student to use the familiar point and click technique to navigate through the content of the system. A screenshot is shown in Figure 1. The left side pane shows the messages that may be generated by the MOO. The pane on the right shows the content of the exhibit the student is visiting, with hot-links to adjacent exhibits at the bottom.

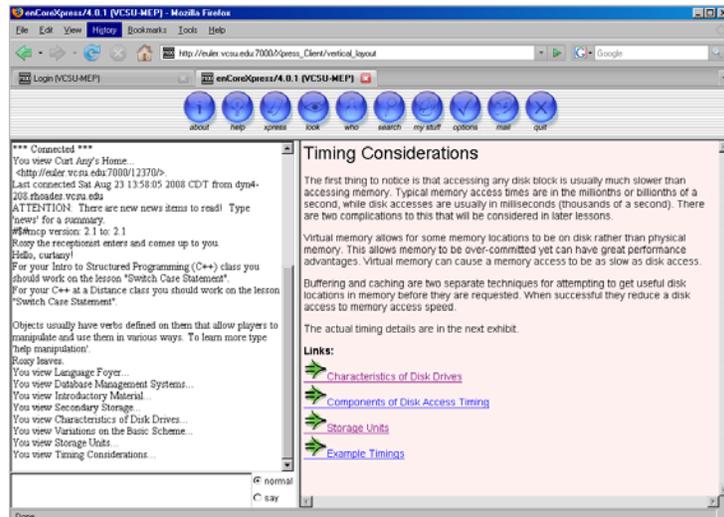


Figure 1: The client display at a particular exhibit.

The MOO contains a variety of interactive objects that the students are to use and any number of these objects might be required for the completion of a lesson. Most of these objects displayed text in the left pane of the client, to which the students tended to pay the least amount of attention. Further, although the client had a graphical user interface it was still graphically portraying text. Since, programming is still mostly a text based activity some of this inevitable, yet it was our desire to make the interface even more engaging.

The MOOApplet[8] was introduced to make the interactions more natural. Any Java applet for which the source code is available may be transformed into a MOOApplet. This applet may possess the normal interaction characteristic of any applet, but it also notifies the system when the student has interacted to a sufficient level to have gained the desired benefit from the applet. Figure 2 shows a user manipulating a binary tree in a MOOApplet.

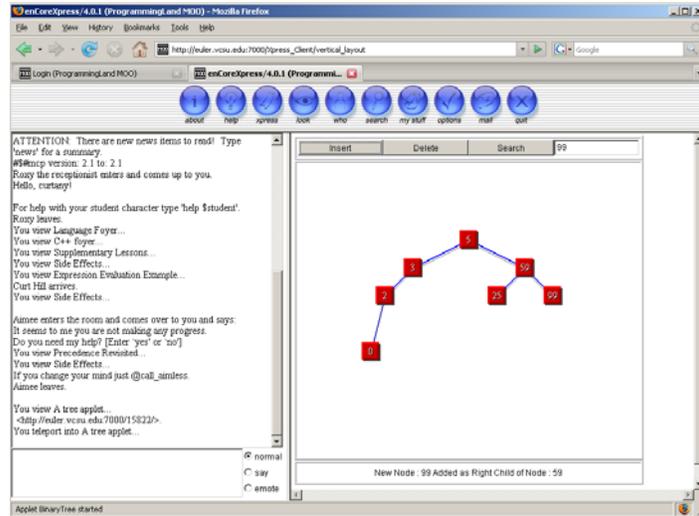


Figure 2: Manipulating a binary tree in an applet.

The MOOApplet could either use the pane that displays content or it could start as a new tab within the browser. This approach was the starting point for the presentation viewer which is discussed in the next section.

## Presentation Viewer

### Design Goals

One of the design goals of the VCSU MEP system is the ability to verify that an objective has been accomplished. This does not guarantee that learning has occurred but learning can not occur if material has not been covered. This approach argues against merely augmenting a presentation with a built in audio lecture and allowing the student to use the Microsoft viewer[12]. There was also no good reason to be tied to a particular proprietary format. Therefore, the viewer was designed to use common graphics, such as JPEGs or GIFs and common audio formats such as WAVs or AIFFs. There will be one graphic for each slide and one corresponding audio file to capture the lecture for that slide. PowerPoint may be used to easily convert a presentation's slides into one of these formats, while numerous sound editors may record the individual commentary on the slide or divide a recorded lecture into the needed smaller files.

Another important consideration is making the requirements of the applet as light as possible. Although Sun has released Java version 6 some time ago[14], many users are still using version 4. Moreover, both images and sound files tend to consume memory, so the implementation maintains only one image and only one sound file at a time within memory.

## Using the Viewer

A student, while browsing through the system may encounter an exhibit which contains a captured presentation. Figure 3 contains a screen shot of such an exhibit. Exhibits containing interactive objects such as a presentation typically have a small amount of text, since the content material is in the presentation. In Figure 3 the presentation is named Algebra. The student merely clicks this icon and the presentation is started.

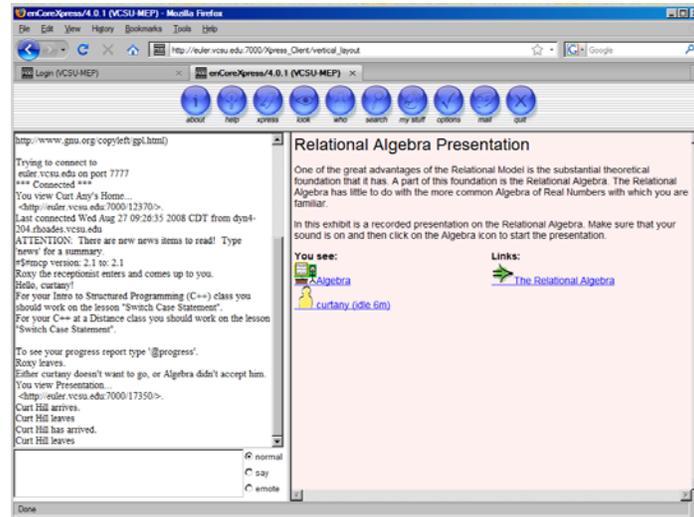


Figure 3: A presentation within an exhibit.

Once the presentation icon is clicked the browser opens a new tab or window and immediately starts the presentation. Figure 4 shows the viewer with a sample slide. The three tabs present in the browser comprise the main user interface. The left most is the login screen. The middle is the current exhibit that the student is viewing, which was shown in Figure 3. The rightmost tab contains the presentation. The student could suspend the presentation and move on to another exhibit by clicking the middle tab. A student may leave a presentation altogether and return to it later. When a presentation starts, the first slide is displayed and the audio lecture is started. For the duration of the first slide the student has few choices. The forward button only works after the lecture portion has completed. The stop button does work but the student does not receive credit for the slide until the lecture segment for that slide has completed. Should a student press stop, the entire slide segment must be re-played.

In the general case, a button's action's enforces the general strategy of the system to be able to verify that material has been completed. The back button moves the student to the previous slide but never replays a lecture segment. The forward button has two different actions, based upon context. If the student is in the last completed slide, the forward button displays the next slide and starts the lecture. If this is not the case, the next slide is displayed but the lecture is not started. To hear a lecture segment a second time requires the play button to be clicked. Thus students may pick and choose which slides to view and listen to a second time. The "play all" button causes the next slide to be started as

soon as the current lecture, followed by a short delay, has completed. The “outline” will display an outline in another window or tab.

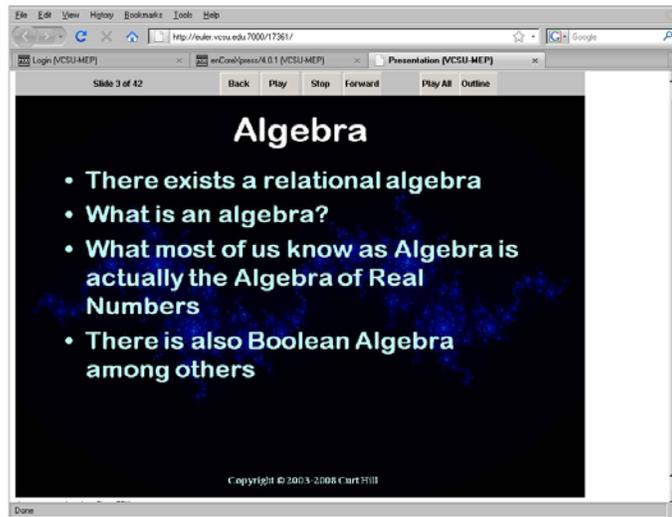


Figure 4: A slide in the presentation viewer.

The viewer always displays the current slide and number of slides present as a convenience to the student. The player is unable to calculate the timings of the lecture segments, so the number of slides is the only indication of the progress or length of the un-viewed presentation.

## Viewer Implementation

The student will not see several things that are carried out behind the scenes. The viewer communicates with the MEP system to record various accomplishments. These include the completion of an individual slide as well as the entire presentation.

The most important event that is recorded is the completion of the presentation. This occurs when every slide has been displayed and every lecture segment has been fully completed and no sooner. This is an event in which the system is very interested, since a system lesson may have this event as a requirement. Thus a student may not be able to receive credit for the lesson until this event is recorded. This system of lessons, requirements and events is discussed at length elsewhere[8].

Normally a student would view an entire presentation in one sitting. They would then receive credit for completion of the presentation. This does not preclude subsequent viewings at any later time, but there is no requirement to do so. Much less frequently, a student may have experienced a part of a presentation, but then have some interruption that prevented completion at that point. Should this student be penalized by being required to hear each of the lectures and view each of the slides as if there was no previous viewing? While it may be good for the student to completely review the

presentation, the applet would be more learner-friendly if it allowed a more convenient restart.

The applet notifies the system upon the completion of each slide and its corresponding lecture segment. The presentation ID and slide number is recorded on the student's MOO object. This information is passed back to the applet when it is restarted. This allows students to be required only to listen to lecture segments that they have not completed.

## **Creating a Presentation**

One of the attractive features of this approach is the presumption that each instructor already has a large number of existing PowerPoint presentations. Thus, there is little consideration given in this paper to the development of the original presentation itself, which is a costly endeavor. One survey found that more than half of experienced respondents took more than three hours to create a half-hour presentation[10].

There are two aspects to the creation of this type of presentation. The first is the creation of an appropriate jar file with the slides and audio. The second is the creation of the MOO object and its placement in the correct room. The former is the more time consuming of these two.

The slides may be exported directly from PowerPoint in either JPEG or GIF format, so this is easily done. In theory any other graphic in this format could become part of this collection of slides, but in practice such a graphic should have been incorporated in the presentation before the conversion. All of the graphics need to be sequentially named with a single format (such as Slide1.jpg). Different presentations may use different formats, but within a single presentation only one format of graphic is allowed. Likewise the naming convention may vary between presentations but not within a presentation. By default, PowerPoint 2007 produces slides with names of the form: Slide1.jpg although GIF files among others may also be easily created. The viewer will properly handle an insignificant zero in this name, such as Slide01.jpg, if it is present.

The sound files are much more labor intensive. There are two approaches, live and studio. In the live approach an instructor records a live presentation to a class. This lecture must then be edited and divided into the individual segments for each of the slides. The studio approach is to create the segments without an audience. Typically the latter approach gives a higher audio quality, however the energy that a teacher demonstrates to a class may be lacking. The viewer also requires that all of the sound files have a common format and naming convention and each slide must correspond to a sound file.

Upon completion of the multimedia component creation, the sound and graphic files need to be placed into a jar file. This encapsulates the presentation into a single file that is placed on the server. Since the slide and sound files typically have non-descript names, this jar file also makes the organization of them on the server much easier by allowing the instructor to give the file a more descriptive name. After this is complete the creation of the object on the MOO will complete the process.

The instructor of the course may create the presentation object within the MOO in the normal way, using the “create” verb. Each presentation object has a setup function to initialize the properties. Two properties must be set, the jar file name and the number of slides. There are several optional properties that the setup will display and set if the defaults are not adequate. These include the prefixes of the names, such as Slide, the formats of the slides and graphics as well as the commentaries that appear above and below the viewer. The defaults are usually correct, but different authors may develop presentations using different software, so some options may be useful.

How have the presentations factored into current usage of the requirements system of the VCSU MEP? Although each instructor may use the system in any way, so far there has been two emergent patterns. The first makes a presentation as an alternative to the normal approach, which is to have a lesson requiring several exhibits that contain text and graphics. A student may examine several rooms of text or view the presentation instead. In fewer cases the presentation is an optional part of a lesson. It is not required, but gives an alternative form of input to the normal text and graphics.

## **Use of the Stand-alone Applet**

The applet may also be presented with a parameter that indicates that it is not connected to the MOO. Doing so loses some valuable features of the applet but allows a wider use. Since the applet cannot record progress on VCSU MEP it no longer prevents a student from hearing later lectures segments. However, the applet may be posted on an ordinary web site.

In the fall semester of 2008, a small study was conducted using an introductory database class. The original PowerPoint presentation was placed on the class web site prior to its presentation. The presentation was then recorded live when it was given. In about three or four days the presentation was removed from the web site and replaced by the applet containing slides and audio.

A survey was given at the conclusion of each test that inquired what sources of information students found to be most useful. The surveys were of no statistical value since only seven students both took all of the surveys and completed the course with a

passing grade. However, they did show a few interesting points. First, they confirmed that some students will download a PowerPoint presentation available before it is given, so they must be finding this a useful feature. Second, the perceived value of the recorded presentations increased during the term.

## **Conclusion**

The presentation Java applet and the corresponding MOO objects were implemented too recently for any suitable study to be discussed here. Moreover, they were used at an institution which is small enough to make statistically valid studies extremely difficult.

This disclaimer aside, there is every reasonable expectation that these will produce a positive impact on the usability of the system. This is the first time that any significant audio has been used in MEP. Until this time the system was exclusively visual, consisting only of text and graphics. The use of audio in these presentations cannot but help those with some kinds of learning styles, who may have been previously discriminated against by the nature of the MOO.

One problem is the lack of interactivity between the student and instructor, in either VCSU MEP or in stand-alone mode. This is something of a problem with computer-generated slides in general, but it is inherently worse in all purely online forms of learning. We are considering the possibility of some type of question mechanism directly in the viewer, where the question is sent to the instructor along with the context of the slide which prompted it. Currently the student may suspend the presentation and send email. The question mechanism would only produce somewhat better integration.

## **Acknowledgments**

The authors wish to thank Martin Kelly, Brad Vender, Otto Borchert and Guy Hokanson for their helpful comments on this paper. We also wish to thank the many students who have used the system and given both positive and negative feedback.

ProgrammingLand was supported by National Science Foundation Grant EIA-0313154 and ND-EPSCoR through the FLARE program under EPS-9874802. The authors gratefully thank and acknowledge both of these organizations.

## **References**

- [1] Belleville, Cathleen. 2000. PowerPoint Historical Review. [http://www.bitbetter.com/downloads/belleville\\_ppthistory.ppt](http://www.bitbetter.com/downloads/belleville_ppthistory.ppt) Date accessed 19 January 2009.
- [2] Boroni, Christopher M., Frances W. Goosey, Michael T. Grinder, and Rockford J. Ross. Engaging Students with Active Learning Resources: Hypertextbooks for the Web. In *Proceedings of the 32nd ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001)*, Charlotte, North Carolina (2001), ACM Press, New York, pp. 65-69.
- [3] Butler, Janet B. and R. David Mautz, Jr. Multimedia Presentations and Learning: A Laboratory Experiment. *Issues in Accounting Education*. Vol. 11, No. 2. Fall 1996.
- [4] Curtis, Pavel, Mudding: Social Phenomena in Text-Based Virtual Realities. *Proceedings of the Conference on Directions and Implications of Advanced Computing* (sponsored by Computer Professionals for Social Responsibility)
- [5] Cyphert, Dale. 2004. The Problem of PowerPoint: Visual Aid or Visual Rhetoric? *Business Communication Quarterly*. March 2004.
- [6] DuFrene, Debby D. and Carol M. Lehman. 2004. Concept, Content, Construction, and Contingencies: Getting the Horse before the PowerPoint Cart. *Business Communication Quarterly*. March 2004.
- [7] Hill, Curt. 2008. VCSU MEP. <http://euler.vcsu.edu/pland.htm> Date accessed 19 January 2009.
- [8] Hill, Curt, Brian M. Slator and Vijayakumar Shanmugasundaram. 2007. ProgrammingLand: A Visualization Enhanced Hypertextbook. Frontiers in Education Conference (FIE 2007). Milwaukee, WI, USA. 10-13 October 2007.
- [9] Hill, Curt, Vijayakumar Shanmugasundaram and Martina Miteva. Use of Agents in ProgrammingLand. In Proceedings of The Eleventh Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006). University of Bologna, Italy 26-28 June 2006.
- [10] Hill, Julie. 2005. More Slides, Less Time, Less Satisfaction. *Presentations*. Vol. 19, Issue 6, p. 7, June 2005.
- [11] Levasseur, David G. and Kanan Sawyer. Pedagogy Meets PowerPoint: A Research Review of the Effects of Computer-Generated Slides in the Classroom. *The Review of Communication*. Vol. 6, No. 1-2, January-April 2006, pp. 101-123.
- [12] Microsoft. 2007. PowerPoint 2007 Viewer. <http://www.microsoft.com/downloads/details.aspx?familyid=048DC840-14E1-467D-8DCA-19D2A8FD7485&displaylang=en> Date accessed 19 January 2009.
- [13] Stevenson, Jerry. 2003. A short history of PowerPoint. [http://stevensonconsulting.com/index.php?option=com\\_content&task=view&id=50&Itemid=2](http://stevensonconsulting.com/index.php?option=com_content&task=view&id=50&Itemid=2) Date accessed 19 January 2009.
- [14] Sun. Java SE Downloads. <http://java.sun.com/javase/downloads/index.jsp> Date accessed 19 January 2009.
- [15] Tufte, Edward R. 2003. The Cognitive Style of PowerPoint: Pitching out corrupts within. Cheshire, CT: Graphics Press.

- [16] Tufte, Edward R. 2003. PowerPoint is Evil. Power corrupts. PowerPoint Corrupts Absolutely. *Wired*. <http://www.wired.com/wired/archive/11.09/ppt2.html> Date accessed 19 January 2009.
- [17] Wikipedia. 2008. Edward Tufte. [http://en.wikipedia.org/wiki/Edward\\_Tufte](http://en.wikipedia.org/wiki/Edward_Tufte). Date accessed 19 January 2009.