1 INTRODUCTION

In this paper we describe a cross-platform, multimedia courseware presentation system developed as a natural evolution of our existing technologies that deliver course content to on-campus and distance learning students. Following more than a decade of developing and deploying web-based education (WBE) systems to support on-campus and distance education students, a number of factors – the need to support a variety of platforms, the increasing availability of high-bandwidth network access, new pedagogies for teaching and learning, etc. – led us to consider a completely new approach to content capture, authoring and delivery. In designing these new delivery and capture systems, we were committed to maintaining our tradition of creating systems that place few demands on instructors and their choice of teaching styles and pedagogy, that provide mechanisms for supporting cooperative and inquiry-based learning [Edelson99], and that recognize the increasing role that content capture and delivery plays in teaching and learning for on-campus as well as distance education. Our goal is to create an open-source, platform-independent and effective content delivery system that retains the functionality from earlier technologies that had proved valuable (though extensive assessment and evaluation) to students and instructors in a variety of teaching and learning scenarios. We also want to capture content in classrooms that are moving from traditional lecture formats to active and collaborative learning where a variety of technologies and pedagogical approaches are employed. As a result, we created a pure Java content delivery system that uses a "plug-in" architecture to facilitate incorporation of tools for annotation, notation, collaboration and navigation; and a capture system that acquires "significant" events, in the form of video keyframes, from unconstrained computer-based

presentations, video streams of instructors, students and whiteboard use, and other sources.

When we began to develop multimedia WBE systems in the mid-1990s, multimedia on the web was still in its infancy. Our project grew from interest by local networking researchers in multimedia protocol performance and design [Padhye98], and it was only the need for experimental data that caused us to deploy courseware for wide spread use in distance education. Initially, a few on-line courses were authored from videotaped lectures and provided free on the Web. These courses were extremely popular and were viewed in over 100 countries by thousands of users. Following this success, we began to support a traditional video-based distance education program, providing on-line and CD-based courseware for over 30 graduate-level engineering, mathematics, and computer science courses.

We began with a fairly simple system that delivered "course slides" in GIF format synchronized with a Real Audio server. Our content delivery system became more sophisticated over time, e.g., better GUI, indexing, search, etc. but required substantial post-capture production work on the text and graphics and on the combined multimedia courseware to improve the quality of the presentations.

As the audience changed from many widely dispersed and "expert" users viewing a few "free" networking courses to hundreds of students enrolled in distance education courses, we found it necessary to create a robust and reliable delivery system, to improve post-capture production time, and to reduce user technical support. For these reasons, coupled with bandwidth issues at the campus gateway and especially for our distance education students located worldwide, we moved to a hybrid delivery system that served

high-resolution content from a CD (or from downloaded files) and used web connectivity for content updates (and later for enhanced functionality). To reduce development costs and to ensure compatibility with the most student users, we used Commercial-Off-The-Shelf (COTS) technologies where we could and standardized on the Windows platform. To avoid requiring instructors to modify their instructional approaches and to ensure quality, we continued to manually author course content. Using various tools to support authoring, we typically were able to create an hour of course content in 3-5 hours of production.

One of the primary limitations of our CD-based hybrid delivery system is that it is bound to the Windows platform and to proprietary file formats such as VML and RealMedia. As bandwidth became less expensive and more available to our user base, and with the expanding use of Linux and Mac OSX, we began to consider cross-platform solutions and a return to fully web-based delivery. Among the many advantages of web-based delivery are the decrease in time lag between when the course is taught on campus and when it is made available to distance students, reduced media and shipping costs, the ability to more easily update content, etc. In 2004, development began on a cross-platform multimedia courseware presentation system, jMANIC – a Java implementation of the MANIC[1] framework. While we encountered some serious challenges, including cross-platform issues and multimedia problems, jMANIC ultimately broke down the platform boundary and became accessible to a wider range of users.

In this paper, we describe the development and application of jMANIC. In Section 2, we provide some general background on web-based education systems, with an emphasis

[1] Java-enabled MANIC is derived from the initial Multimedia Asynchronous Networked Individualized Courseware (MANIC) project begun in 1995 [Stern97b].

on related work on "electronic lecture" (often called "record-and-playback") formats. In Section 3, we describe the high-level jMANIC architecture. In Section 4, we give details on the functions provided by jMANIC that were adapted from or based on CD-MANIC extensions. In Section 5, we provide examples of jMANIC applications. In Section 6, we discuss some related work on automatic capture and production and in Section 7 we describe challenges and future work.

2 BACKGROUND

In this section, we describe the evolution on the MANIC technologies that most directly influenced the jMANIC design and development. We begin with a brief overview of related work.

2.1 A Taxonomy of Web-Based Educational Systems

For context, it is useful to place our project within a taxonomy of Web-based education (WBE) systems. Brusilovsky and Miller [Brusil03] describe five main aspects of WBE systems: structure; type of content material; media; authoring; and delivery. They divide structure into electronic textbooks and electronic presentations, and further divide presentations into lectures and guided tours. Media and content type are closely related – text, static and dynamic graphics, simulations, audio and video supported by HTML, multimedia, animation and graphical formats, and audio/video encoding and delivery.

WBE systems are authored with simple text editors (some with HTML conversion), general tools (e.g., GoLive, ColdFusion, etc.) and special educational tools (such as structured content authoring tools). Electronic lectures combine text, graphics, images, and animations (in various formats) with audio/video. The audio/video can be captured and automatically synchronized with the text/graphics/images/animations or converted

from another recording medium. Similarly, the text/graphics/images/animations can be captured or separately authored.

Content is delivered using technologies ranging from simple static HTML to scripted navigation to full dynamic web services with database support. Electronic lecture content can be delivered synchronously or asynchronously. Brusilovsky and Miller distinguish asynchronous lectures by the level of "chunking" – from a whole lecture to segments as fine as a single line of text. The level of chunking is directly related to index- and search-based navigation. A whole lecture "chunk" is essentially a web-based video without any easy means for navigation other than sequential viewing. Almost all current systems support navigation at the "slide-level" through VCR-like controls. Many have some form of course index, often a text table-of-content for manually authored or PowerPoint®-based presentations and typically video thumbnails for automatically captured presentations. In some rare cases, tables-of-content are generated from sophisticated OCR, speech, or hand writing analysis. In addition to course indexes, some systems support search – again typically over text in slides or slide titles, but occasionally using speech analysis. A number of record-and-playback systems provide dynamic screen capture from PC-based presentations, as we describe in more detail below.

Brusilovsky and Miller [Brusil01] note that web lectures "[are] becoming more and more popular [for] presenting course material on the Web … [are] still the best replacement for classroom lectures [better than handouts or textbooks] … [and are] the easiest way to place some course content on the Web." They may be as effective as in-class lectures [LaRose97], particularly in large lectures. In contrast, the President's Information Technology Advisory Committee [PITAC] notes that while record and

playback technologies are "useful and likely to be profitable ... these are near-term and transitional means for e-learning." However, Fletcher and Dodds [Fletcher00] still found a 0.5-sigma improvement with interactive multimedia. Harley, et al [Harley02] reported a substantial increase in student satisfaction when archived multimedia lecture material was made available. Shih [Shih01] found substantial improvement in student perception of learning and satisfaction over a variety of technologies. We have shown [Burleson02a] that record-and-playback approaches are effective (particularly when coupled with appropriate pedagogy [Adrion00]) and they clearly address the issue of rapid content creation.

2.2 The Evolution of MANIC

The initial Multimedia Asynchronous Networked Individualized Courseware (MANIC) [Stern97a, Stern97b] prototype was developed to deliver online, asynchronous distance education courses associated with a traditional Video Instructional Program (VIP) that served the College of Engineering and the National Technological University (NTU)[2]. Course content was authored from the videotapes (and later, DVDs) produced by VIP. The system employed a client-server model and delivered a set of lectures for each on-line class. Each lecture consisted of a set of HTML slides linked together and synchronized with an audio stream of the original lecture. The server-side consisted of a streaming audio server[3] coupled with a set of Perl scripts invoked through a web server; the client-side consisted of HTML and Javascript enabled controls, synchronized with streaming audio received using a multimedia plug-in contained within a web browser.

[2] NTU [http://www.ntu.edu/] was purchased by Sylvan Learning Systems (now Laureate Education Inc.) and became part of Walden University; the VIP program was merged into the campus Continuing Education programs on June 1, 2006.
[3] The Real Audio server was used.

Each on-line course had a slide index and a simple Boolean search function. The reliance on the proprietary Real-Player plug-in proved to be a major disadvantage that caused a significant number of requests for technical help. Revisions of the plug-in were frequent causing MANIC to cease functioning and the free version of the plug-in was while publicly available was well "hidden" on Real Media's frequently redesigned website.

To address browser/plug-in compatibility, platform (OS), and network/bandwidth issues, a new version of MANIC was developed as a Java applet. Unfortunately, this attempt was premature. At the time there was a wide variance in Java virtual machines on different operating systems and all individual user machines. This version was replaced with DB-MANIC [Schapira01], which organized course data within an SQL database, was driven by PHP scripts on the server side, and used Javascript and a multimedia plug-in on the client side similar to the original version.

In the Brusilovsky-Miller taxonomy, our first generation systems were asynchronous electronic lectures, manually authored using both general and special tools and delivered using scripting languages and commercial browser plug-ins. Chunking was at the slide level, although we did experiment with finer grain chunking [Stern98, Stern00]. Representative contemporary record and playback technologies include: the MIT Physics Interactive Video Tutor project (PIVOT) [Lipson01], the Georgia Tech Classroom 2000 (now eClass) [Abowd96, Abowd99], the CMU Just-In-Time system [Danne97, JITL], the CCNY Virtualized Classroom [Zhu04], Multimedia Asynchronous Networked Individualized Courseware (MANIC) [Stern97a, Stern97b], e-seminar [Steinm01], Project Zeno [Mukhop99], Authoring-on-the-Fly [Bacher96], WLS [Bacher96], Syncomatic [Severance00], Dublin Virtual Lectures [Smeaton97] and IRI-h [Maly01].

In the late 1990s, we sought to include video content in MANIC but were faced with bandwidth limitations and firewall issues at student sites. In 2000, CD-MANIC was devised as the next generation of software for multimedia courseware delivery. CD-MANIC employed a hybrid network model, where it accessed course content, such as slides and videos, locally and it used the network to check for updated course content and to provide other functionality. CD-MANIC was designed to display slides as either HTML or PowerPoint® exported to Microsoft's Vector Markup Language (VML). CD-MANIC used the proprietary RealMedia® video format, because it rendered high quality video that was highly compressed – a full semester course could be stored on either 5 or 6 CDs, or a single DVD. At the time when CD-MANIC was being developed, other video codecs were being considered, including Quicktime, Windows Media, and MPEG. The RealMedia® format was by far the best due to its quality, compression, and the availability of tools for production and maintenance of videos. The one major drawback of the RealMedia® format is that it is a proprietary format.

CD-MANIC proved to be a stable and inexpensive content delivery system and was successfully deployed to support more than 30 graduate courses within the VIP/NTU programs and was also used at a number of other universities, including Polytechnic University of New York and the North Carolina State System. Because of it uses a browser and plug-ins prepackaged on the CD, the system requires a much lower level of technical support, and allows users to get access to course content regardless of whether or not a connection to the Internet is available.

CD-MANIC might not be considered a WBE system, but it is a hybrid system that delivered content from the user's PC (installed from the CD/DVD or downloaded). While

initially network connectivity was limited to providing updates, several CD-MANIC extensions made significant use of the Web. We linked a multimedia MANIC presentation with an online textbook. A student had access to both the book and the course lecture indexes, and could use either to navigate within the text or lecture (and quickly flip between the two). The Learner Logger [Burleson02b] captured all student interactions with the CD-MANIC system (with permission) and uploaded these to the server whenever network connectivity was available. A report generation system analyzed student data and was extremely useful as an assessment tool – showing where students spent the most time, how students navigated the course content, etc. These analyses could be correlated with student performance and were valuable to the instructors, the course designers and the MANIC development team. The extended search mechanism [Kumela04] took student queries, found the best matches within the course content, and created a Google® query from these matches that returned related material from the web. CLIMANIC [Ray04] allowed multiple CD-MANIC users to share controls over the web, while content was delivered from each user's PC, and to carry on a live chat while viewing and navigating the content. The notation system provides both private notes and the ability to publish the notes for viewing by all users. Public notes are automatically updated and shown as an alternative index (to the Table-of-Contents index) into the course content.

While content delivery is not web-based (except for updates) in CD-MANIC, it does support both synchronous and asynchronous interaction and provides a number of important tools for notation, annotation, navigation and collaboration. Our challenge in developing jMANIC was to adapt as much of the CD-MANIC functionality as possible

while addressing platform interoperability. Our solution, as described below, was to develop a plug-in architecture to allow the easy and independent development of tools for jMANIC.

2.3 Content capture and production

Automated production is a major challenge for record-and-playback technologies. While one would like to post courseware in real-time or at least within a short time following a captured lecture, quality (as measured by the ease of navigation, the availability of search, the quality of the "slides," etc.) typically suffers in many of the "automatic capture" systems. Many systems require significant manual effort in analog data collection, digitization, and synchronization. In our courseware production, we emphasize low demand on instructors and high quality presentations. As a result, it can take three to five hours of labor of a trained courseware developer to create a one-hour lecture. Some systems have introduced forms of automatic production. Cornell's lecture browser [Mukhop99] includes lecturer tracking, slide change detection and segmentation, and matching slide projections with digital slides. A precursor to Cornell's lecture browser, UC Berkeley BMRC Lecture Browser [BMRC] has several additional features including audio search capabilities, bookmarks and an interactive whiteboard for online students. Microsoft is a leader in seminar and lecture recording, with over 6000 videos recorded using a "virtual director" to replace human operators [Liu01, Rui01, Rui03, Rui04]. In these videos, an index is generated from PowerPoint® slides that accompanied the talks. This approach also depends on a seminar format, incorporating slide presentations, in order to create a visual record. The eClass system [Abowd99, Brotherton98] allows an instructor to use either an electronic white board or a PC. The

lecturer can annotate any slide they choose and use any program on the eClass classroom computer. The system is able to create an index from notes on the electronic white board, URLs visited, and slides. It stores all of these, including slides with annotation, in a manner that is transferred to web pages for later review by students. Mediasite [Mediasite] has various systems that will automatically record a lecture and create a visual thumbnail index. The system also allows a lecturer to add annotations as part of the visual index. Mediasite's algorithms determine which frames to capture and store, typically storing a frame every time a change occurs.

To enable automated production, we need a way to capture many classroom activities. MIT Media Lab's SmartCam [Pinhanez95] has a robotic TV camera without a human operator, changing its attitude, zoom and position to provide specific images upon verbal request from a director. The Intelligent Classroom [Franklin99, Flachsbart00] uses cameras and microphones to determine what the speaker is trying to do and then takes the actions (e.g., jumping to a new slide) it deems appropriate. Auto-Auditorium [Cruz94, Branchi98] uses multiple cameras that automatically switch based on context**.** The automatic camera management capability (iCam) of Microsoft Research [Rui01, Liu01] automatically tracks both the speaker and the audience in a lecture room environment by using computer-vision- and microphone-array-based techniques. The BlueEyes project of IBM Almaden Research Center [IBM-BE] uses non-obtrusive sensing technology, such as video cameras and microphones, to identify and observe a user's actions, and to extract key information, such as where the user is looking and what the user is saying verbally and with gestures.

3 JMANIC ARCHITECTURE

The jMANIC content delivery system maintains the same Graphic User Interface (GUI) as its immediate predecessor, CD-MANIC [Thampuran01, Thampuran02]. The application plays back synchronized media content (i.e., lecture video and slides) and allows the user to quickly navigate through the lecture via the slide index component. It differs from CD-MANIC in that it is a pure Java implementation of the MANIC engine, and is therefore platform independent. Moreover, it can play back content stored on the web as well as locally stored media. jMANIC also features an extensive plug-in architecture, allowing for rapid plug-in development, which makes the courseware highly customizable.

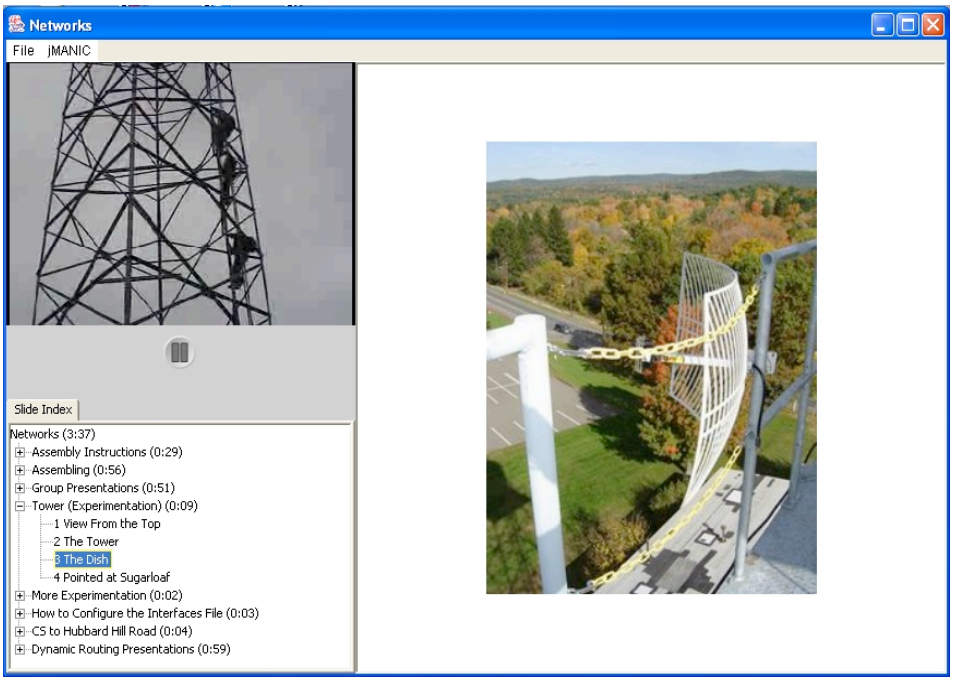

FIGURE 1: THE JMANIC USER INTERFACE

A difficult issue associated with cross-platform media application development is video support in Java. Neither the Java Media Framework [JMF] nor QuickTime For Java [QTJava] support video codecs with satisfactory quality and compression rates for the purposes of streaming lecture video. The default video component in jMANIC uses the

IBM Toolkit for MPEG-4 [IBMTK] to achieve cross-platform playback of MPEG-4 encoded video. This library also supports playback of streaming flavors of MPEG compressed videos streamed via HTTP, allowing jMANIC to open either local or remote video files. This is vital to the system architecture because an important attribute of jMANIC is its ability to access content over the web.

Previous versions of the MANIC engine faced compatibility issues; in the case of DB-MANIC, high bandwidth requirements meant end-users without a broadband connection were effectively unable to access the system. CD-MANIC addressed this issue by playing back locally stored content, but was restricted to the Windows 32-bit platform. We wanted to create a new, flexible, cross-platform implementation of the MANIC engine capable of handling different multimedia types, both locally and remotely stored. To this end, the jMANIC high-level design does not commit itself to any specific media type or platform. Further, jMANIC can playback both local and remote content, and can be launched locally or from the web via Java WebStart®. To achieve this flexibility, jMANIC was built in pure Java, in a modular fashion that defers explicit representation of specific media types (e.g., the video or slide format) to subclasses.

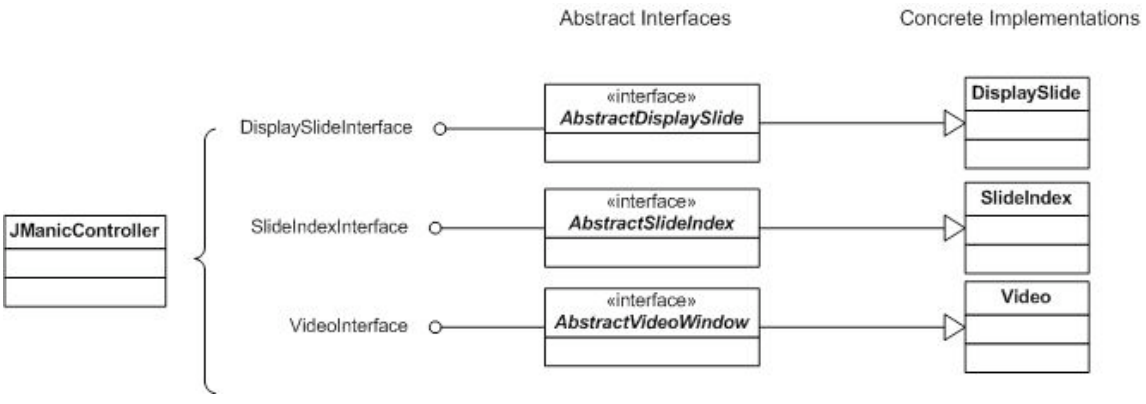

FIGURE 2: SCHEMATIC DEPICTION OF JMANIC ARCHITECTURE

There are three primary components that compose the jMANIC GUI; the video display, the slide display and the slide index. Each of these components is associated with an abstract type; this exploits inheritance in order to decouple implementations of the different components from their respective interfaces. In order to avoid explicit communication between all of the respective components, jMANIC employs the **Mediator** design pattern [Gamma95]. The `JManicController` class holds pointers to the three main component types, and each component holds a reference to the controller class; all communication between these components is done via the abstract class interfaces through the mediating controller class. This provides a layer of abstraction between the concrete implementations and the abstract objects that the controller interacts with. Moreover, it abstracts away how the different GUI components interact with one another. In turn, this allows for the ability to support new media types by sub classing the desired component and simply plugging the new implementation into the system. For instance, in order to support a slide type other than the default format, which is SVG (Scalable Vector Graphics), one can subclass the `DisplaySlide` abstract super-class, adding the functionality to support the desired media type, and then plug it into the system at runtime via the provided plug-in architecture. This means that one needn't re-compile the jMANIC application in order to swap out any of the three primary components; it can be done at runtime through the interface. This kind of flexibility with respect to media formats is especially important in order to accommodate disparate teaching styles as well as emerging content delivery formats.

3.1 A Plug-In Architecture

We found that creating plug-ins for courseware is a vitally important tool for achieving collaborative and constructive virtual learning environments. For example, the CLIMANIC [Ray04] plug-in was developed for CD-MANIC, which enabled students and professors to collaborate via the web and participate in virtual 'sessions'. However, writing plug-ins for CD-MANIC involved directly modifying the MANIC engine's code, and thus developing these tools took a substantial amount of time and expertise. In jMANIC we address this issue by providing easy extensibility via the plug-in interface, which provides an Application Program Interface (API) to the MANIC engine and tools to decorate the jMANIC GUI.

The plug-in architecture supports dynamic loading of classes that implement the `AbstractJManicInterface` super-class at run-time. Moreover, these plug-in classes can be loaded either remotely or locally; that is, jMANIC is capable of loading up a plug-in JAR file sitting somewhere on the web. At launch, the application parses an XML (eXtensible Markup Language) file that contains the data needed for jMANIC to open a presentation; this file includes an optional path to a remote directory containing plug-ins. Note that this means that plug-ins can be 'released' after a course has been distributed, without any change on the users end, so long as the client jMANIC application is pointed at the correct remote plug-ins directory via the aforementioned XML file.

4 JMANIC EXTENSIONS

The plug-in architecture allows for rapid development of new extensions for jMANIC. CD-MANIC featured a Google™ enhanced search engine (see Figure 3) that used the Lucene search API [Kumela04]. In order to integrate this search with jMANIC; an

adapter class was introduced to wrap the search in an interface compatible with the jMANIC plug-in architecture. This is demonstrative of the ease with which new plug-ins can be developed for jMANIC.

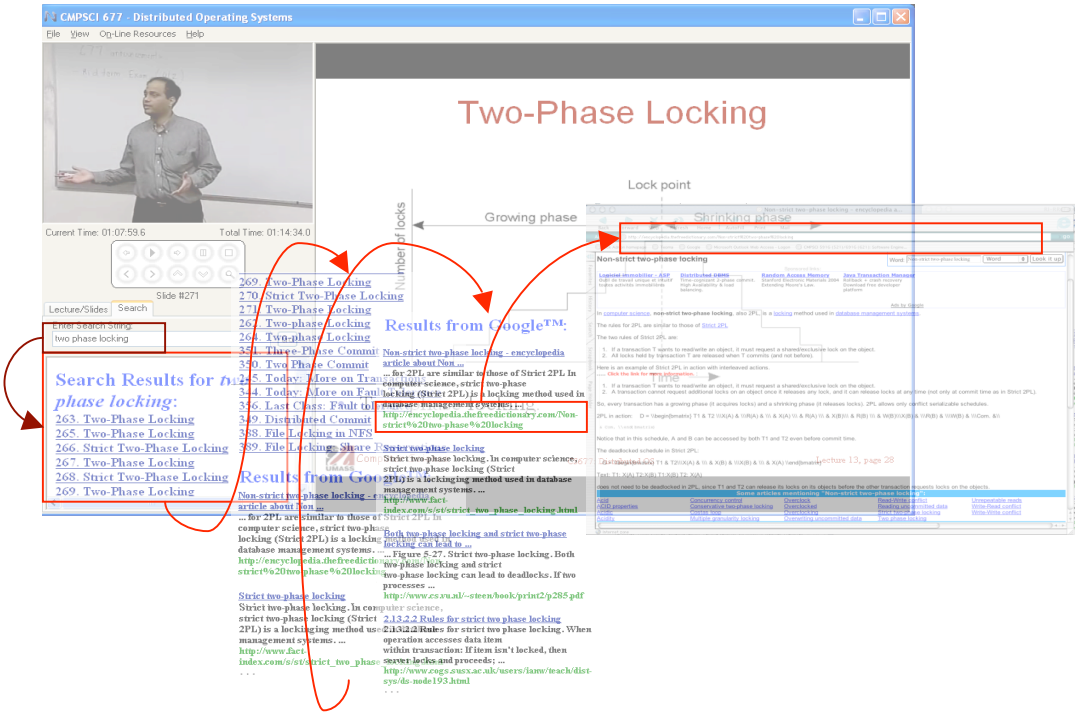

FIGURE 3. EXTENDED SEARCH EXAMPLE

Another extension that has been developed for jMANIC is a note-taking system. The plug-in allows users to record (text) notes that are associated with a particular course and slide. As the lecture progresses, any recorded notes taken for the current slide are displayed to the user. Notes can be made either private (viewable only by the user that entered them) or public (visible to anyone accessing the same course). Moreover, the plug-in allows users to add useful links to their notes. The benefits of this plug-in are two-fold. It fosters an individualized courseware experience by allowing the user to add personalized notes on specific slides and it also encourages collaboration by enabling users to makes notes public, for all users to view.

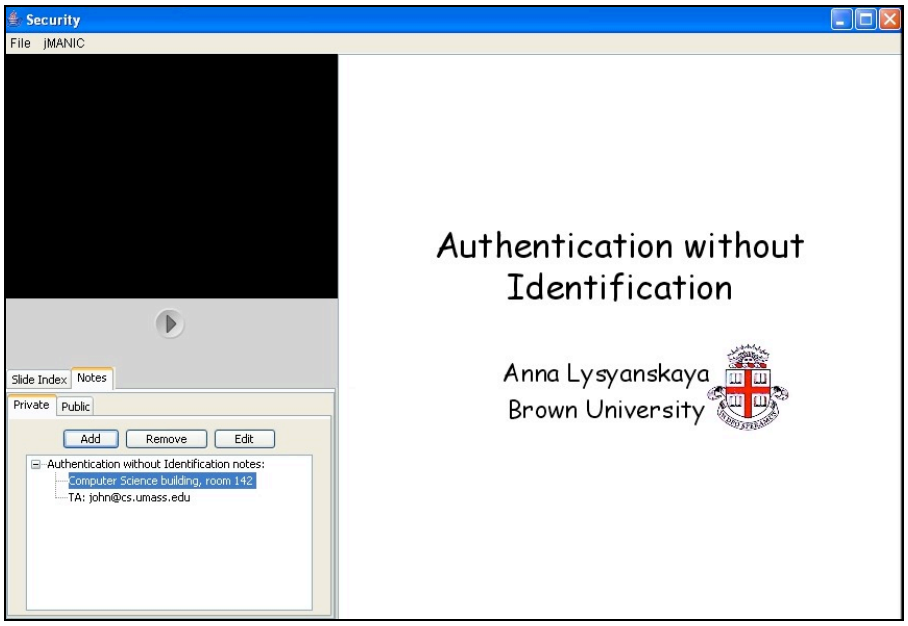

FIGURE 4: JMANIC WITH NOTE-TAKING PLUG-IN.

A third extension for jMANIC that is currently in development is a collaborative annotation tool, called 'jScribble'. This module is a decorator for the `AbstractDisplaySlide` component and plugs into the GUI as the slide window. It enables users to mark up slides with editing tools including a pencil, paintbrush, highlighter and text box. The program then uploads these user mark-ups to a server. The markups are then (optionally) displayed within other jMANIC clients when viewing the same class.

The jScribble tool is inherently collaborative, allowing students in a class to edit content and also to view slides edited by other students. By default, slides are edited and viewed in an asynchronous fashion; students view lectures and annotate the slides individually at different times. There is also a synchronous mode, which allows for real-time collaboration and chat between students in the same class that are logged on at the same time. Users can query to see if there are any open 'sessions' for the course that they are currently accessing. Every session has a group leader (group leadership can be

requested by any member of the session); the leader is the only participant who can mark up slides during a session. We envision a professor using this tool to hold 'virtual office hours', allowing students to log onto a session to conduct group problem solving.

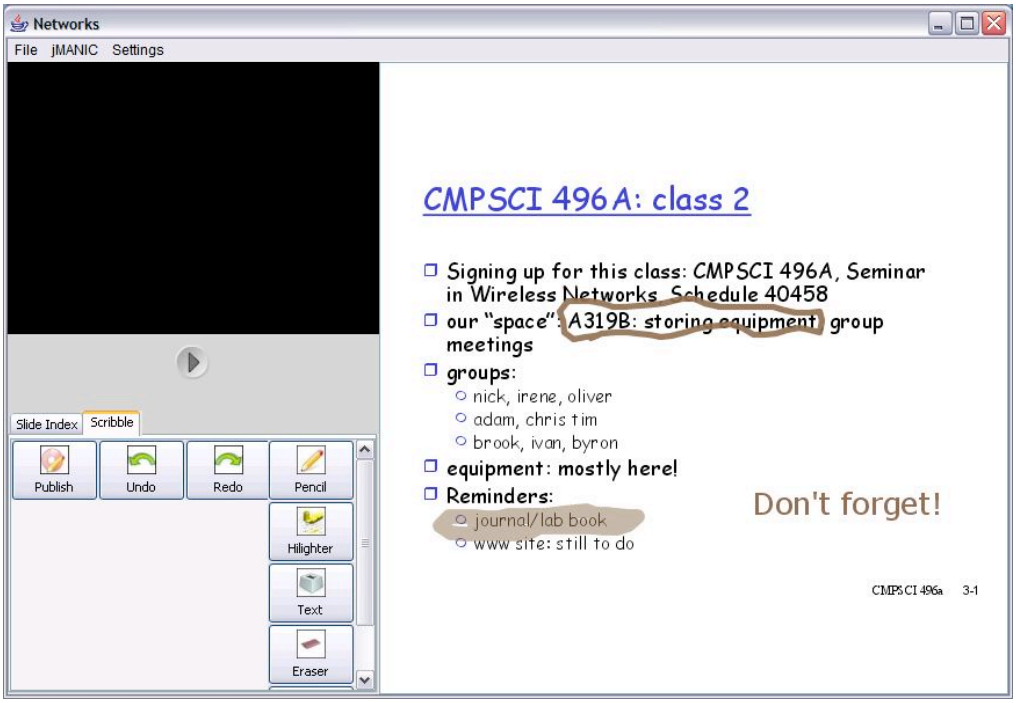

FIGURE 5: JMANIC WITH 'JSCRIBBLE' ANNOTATION PLUG-IN.

This tool is somewhat similar to the Classroom Presenter [Anderson04] tool developed at University of Washington, but differs in that jScribble was developed as a collaborative tool to be used with archived content for distance learning as opposed to a distributed real-time lecturing tool.

Search, the Note-Taking plug-in and jScribble are all examples of extending courseware in order to provide a more constructive, collaborative and personalized learning experience. The flexibility of the jMANIC plug-in architecture allows for rapid-prototyping and development of add-ons for extending basic record-and-playback technology.

5 JMANIC APPLICATIONS

While still a relatively new piece of software, jMANIC was first used in the fall of 2005 to create a course archive and online lab notebook for Computer Science 496A, "Wireless Networks". Together, Professor James Kurose and research scientist Michael Zink led a team of 9 (mostly undergraduate) students to develop, deploy and measure novel outdoor networks based on the 802.11 standard. The course had as its aim the measurement of link throughput over long distances using 802.11 together with directional antennas. This course was unique in that the research conducted, as part of the course, in addition to being useful to the students, was directly applicable to research being done within the National Science Foundation funded Engineering Research Center, CASA (Collaborative Adaptive Sensing of the Atmosphere) housed at UMass.

For CS 496a, jMANIC was used as an archival tool for CASA researchers, students and the research community rather than as a means for supplemental self-study. Both the lectures and the experiments were encoded into the jMANIC format and posted to the web for public access. The archived course could then be used both by those in academic settings as well as by the public for purposes of dissemination and outreach through multimedia. A Macromedia Flash© front end was created for the jMANIC presentation and other data (including graphs, images and instructions). The course can be accessed here[4]. Note that Flash is cross-platform (one need only have a Flash player installed) and so this does not compromise the platform independence of the presentation. The front-end, shown in Figure 6, provides access to a collection of content relevant to CS 496a, including research papers, data collected during experiments, pictures, the jMANIC presentations and PowerPoint© slides. [Wallace06]

[4] http://manic.cs.umass.edu/jMANIC/fall05/cs491/WirelessNetworksWeb.html

FIGURE 6. WEB PAGE FROM ONLINE VERSION OF CS496a

In addition to CS 496a, jMANIC was used to produce online versions of lectures given at the University of Massachusetts in 2006 as part of the "Five College Speaker Series on Information Assurance" a series of lectures funded by the National Science Foundation's Scholarship for Service program. The lectures can be accessed at the following address[5]. The ARIA group (Academics and Research in Information Assurance), a research group within the Department of Computer Science at UMass wanted a cross-platform, low-cost way to make the lectures from the Five College Speaker Series publicly available without impinging on the lecturing style of their speakers. Given the goals of jMANIC it fit the needs of this group perfectly.

6 AUTOCAPTURE

Over the years we have experimented with a number of automatic capture systems, both home grown and from other developers. We did not adopt any of these for production use, since most were either expensive and/or required an instructor to take specific

[5] http://aria.cs.umass.edu/speakers/index.html

actions to start and/or compile the lectures. Manual production for those who used whiteboards required us to recreate each entire lecture in PowerPoint or HTML, but few people used whiteboards exclusively. Many more used a whiteboard in conjunction with a computer-based presentation, and the number of instructors using tablet-PCs increased dramatically. Also a large number of instructors demonstrate systems, show simulations, and use applications in the classroom. While we used a number of dynamic screen capture systems [Caffery05], we found the storage requirements and the difficulty of authoring a drawback. Thus, we developed a new approach to automated capture.

A number of systems, as described in Section 2.3, have been developed to automatically capture classroom events [Abowd99, Bianchi98, Franklin98, Machnicki02], and to create various delivery modes (web-based lecture notes, record-and-playback systems). Our system can capture and analyze a variety of classroom events from a number of media streams (PC capture, cameras, other sensors and devices), store these data, and produce meaningful records that can be used by students for study and review, and for distance education.

The system is currently used to sample and analyze the output of any PC (desktop, laptop, tablet with any installed OS and hardware) used by the lecturer; automatically creating a compact visual record in the form of a small set of selected images (key frames) that represent "significant" points in the presentation. Analyzing the sample key frames and determining whether they are in fact significant can measure the effectiveness of the system. However the notion of "significant" is largely qualitative. For the cases where a lecturer is using a PowerPoint® or other slide presentation with no inking, one could define various quantitative measures, such as comparing the captured presentation

with the actual presentation (slide transitions or animations missed or duplicated). In situations with more complex events, e.g., data entry or inking, it is much more difficult to define quantitative measures. Suppose an instructor was underlining words or phases with a tablet stylus for emphasis. One would not want to capture each pixel as it is added, but rather a completed underlining. Similarly, if the instructor is entering lines of code into an editor, one would not want to capture this event pixel by pixel or even character by character. In the end, it is the classroom *context* that matters. Significance is related to selecting images that represent a significant classroom event. In slide presentations; it is adequate to assume a slide transition represents a significant event. With inking and data entry, a word or paragraph, a single or series of annotations, a line of code, etc. might each be significant.

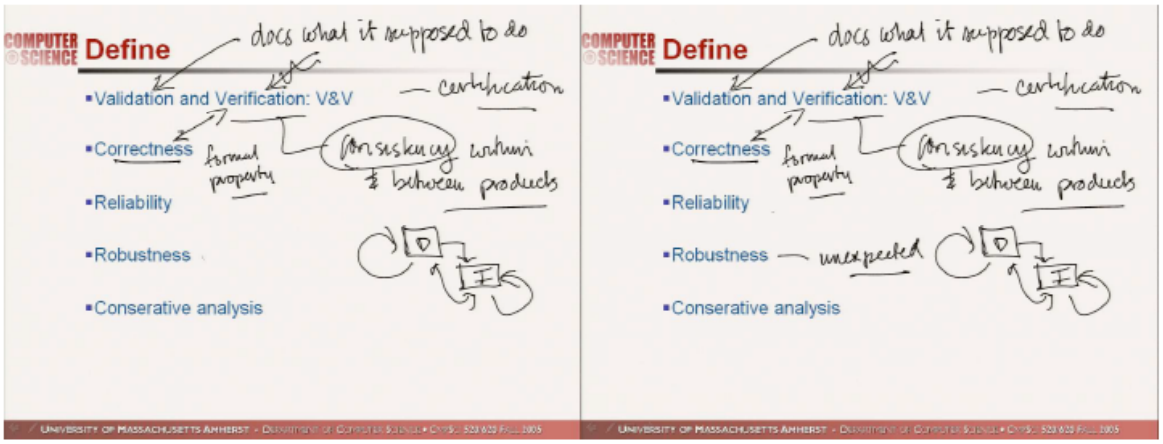

FIGURE 7. TWO CONSECUTIVE CAPTURED KEYFRAMES

The algorithms run on a second PC and analyze data captured by a device that splits the signal going from the lecturer's PC to the projector. The system is platform independent, requires no software to be loaded onto the presentation computer, and requires no prior training or special preparation, making the whole process transparent to the lecturer. Our goal is to combine the capture system with the jMANIC delivery system

to produce jMANIC records of the classroom events within a few minutes of the end of the class period. At present, we can produce the captured events and timing information, but still has to build the jMANIC presentation manually. From a performance perspective, the capture algorithm runs in close to real time, completing all passes within 5 minutes of the end of a 75-minute class.

As an initial data collection experiment, we captured class presentations in three separate computer science courses; the captured video frames consumed between 5 and 6 GB of storage and contained anywhere from 4000 to 21000 images. Our system reduces the tens of thousands of frames captured in each 75 minutes lecture to around 100 key frames that can easily browsed or indexed and searched.

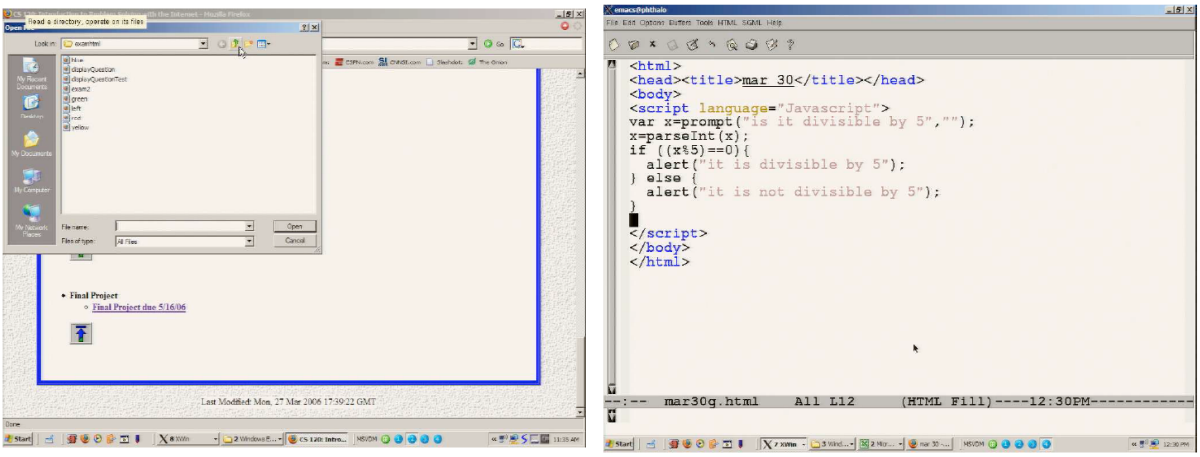

FIGURE 8. CAPTURED KEY FRAME EXAMPLES

Obviously, the system could store every image as a key frame in order to ensure that no significant points in the lecture were skipped. The result would be equivalent to a dynamic screen capture while our goal is to determine a small set of images that represent significant classroom events. Over all of the datasets, the system was able to identify and store 99.8% of the slide changes one normally would consider significant and in which no signal noise was found. The system was able to identify the presence of annotations

and stored key frames including intermediate annotations and the completely annotated slide. The stored intermediate markings adequately represent the lecturer's notational progression as it occurs, i.e. not all markings, but those that correspond to significant classroom events. Where data entry events such as code development and application demonstrations were captured, the sequence of stored images creates an accurate and meaningful record of significant classroom events. For example the pull-down menu example in Figure 8, the algorithm captures persistent events and ignores transitory events.

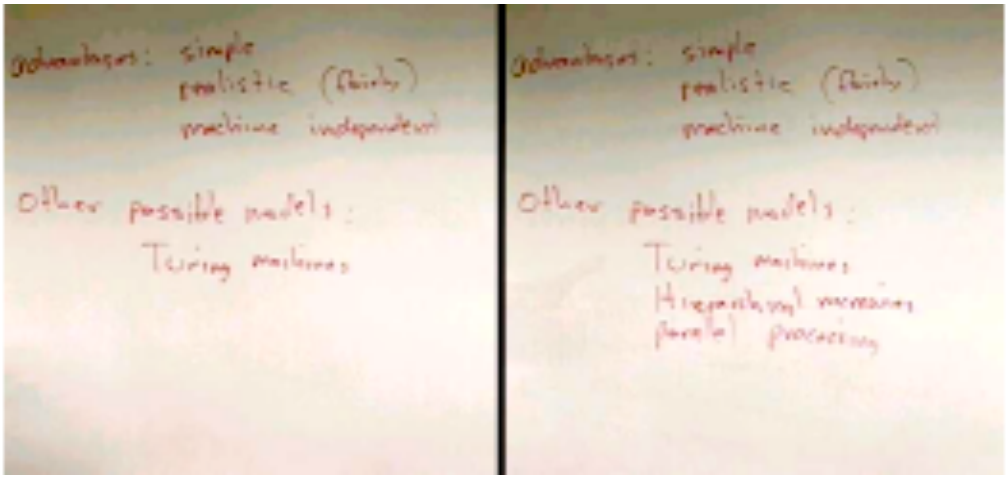

FIGURE 9. TWO SUCCESSIVE CAPTURED KEY FRAMES

Since we are interested in capturing key frames from other sources, we applied the algorithm to a computer playing a DVD of a lecture. While camera movement and zooming by the human operator did affect the algorithm, results are promising. Figure 9 shows two key frames extracted in this experiment.

7 SUMMARY & CONCLUSIONS

Over the years we have adapted the MANIC engine as new technologies have arisen. In the late 90s Internet-based content was largely unfeasible for users without broadband connectivity. The introduction of CD-MANIC for local playback remedied this but

resulted in platform-specific (Windows) courseware. Further, CD-MANIC did not allow optional remote content playback. During the evolution of MANIC we have discovered that building plug-in components for collaborative and customized learning was a necessary tool for providing a useful virtual pedagogic environment.

The latest implementation of the MANIC engine, jMANIC, addresses the Windows-only problem because it is cross-platform. Moreover, it allows for either local or remote content playback, allowing for flexibility in terms of content storage and delivery. We designed jMANIC around a plug-in architecture to allow for rapid plug-in tool development. In addition, jMANIC was successfully used to document the novel undergraduate research-based course Computer Science 496a, which was posted online and also distributed on disk, as well as Information Assurance lectures, which were posted online.

We are in the process of extending the capture system so that camera sources (instructor tracking, white board use) are included and attempting to fully automate the production of jMANIC class records using the captured images, timing information and indexes from OCR analysis.

While jMANIC is the most flexible version of the MANIC engine thus far, some issues remain to be addressed. One such issue is Linux compatibility; jMANIC is known to have trouble running on some flavors of Linux. Another issue is the authoring process, which at the moment is semi-automated. We also plan to develop an export-import tool that will generate jMANIC-ready presentations from other MANIC formats as well as from the autocapture system. This will allow us to import the entire current CD-MANIC inventory of courseware and outreach titles.

8 REFERENCES

[Abowd96] G. D. Abowd, C. G. Atkeson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney, and M. Tani, Teaching and learning as multimedia authoring: the classroom 2000 project. In *Proceedings of the Fourth ACM international Conference on Multimedia* (MULTIMEDIA '96), ACM Press, New York, NY, pp. 187-198. Boston, Massachusetts, United States, November 18 - 22, 1996.

[Abowd99] G. D. Abowd, Classroom 2000: an experiment with the instrumentation of a living educational environment, *IBM Systems Journal*, v.38 n.4, pp. 508-530, Dec. 1999

[Botherton98] Jason A. Brotherton, Janak R. Bhalodia, and Gregory D. Abowd, Automated Capture, Integration, and Visualization of Multiple Media Streams, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp. 54, June 28-July 01, 1998

[Adrion00] W. R. Adrion, Developing and Deploying Software Engineering Courseware in an Adaptable Curriculum Framework. In *Proceedings of ICSE-2000*, Limerick, Ireland, June 2000

[Anderson04] R. Anderson, R. Anderson, B. Simon, S. A. Wolfman, T. VanDeGrift, and K. Yasuhara, Experiences with a tablet PC based lecture presentation system in computer science courses. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (SIGCSE '04), ACM Press, New York, NY, pp. 56-60. Norfolk, Virginia, USA, March 03 - 07, 2004

[Bacher96] C. H. R. Bacher, C. and T. Ottmann, Tools and services for authoring on the fly. In the *Proceedings of the Conference on Educational Multimedia and Hypermedia (ED-MEDIA '96)*. Boston, MA, June 7-12, 1996.

[Bloom84] Benjamin S. Bloom, The 2-sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher,* Vol. 13, No.6, pp. 4-16, 1984

[BMRC] BMRC Lecture Browser http://bmrc.berkeley.edu/frame/projects/lb/ last viewed 5/31/06

[Branchi98] M. H. Bianchi, AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations. In Joint DARPA/NIST Smart Spaces Technology Workshop, 1998. http://www.autoauditorium.com/ last viewed 5/31/06

[Brotherton04] J. A. Brotherton and G. D. Abowd, Lessons learned from eClass: Assessing automated capture and access in the classroom. *ACM Trans. Comput.-Hum. Interact.* Vol. 11, No. 2 pp. 121-155, Jun. 2004

[Brusilovsky01] Peter Brusilovsky and Philip Miller. Course Delivery Systems for the Virtual University. In: F. T. Tschang and T. Della Senta (eds.): *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University.* Amsterdam: Elsevier Science and International Association of Universities, pp. 167-206, 2001

[Brusilovsky03] Peter Brusilovsky and Christoph Peylo, Adaptive and Intelligent Web-based Educational Systems. In *International Journal of Artificial Intelligence in Education* vol.13, pp. 156–169, 2003

[Burleson02a] Wayne Burleson, Stephen Kelley, and Santhosh Thampuran. A new course in multimedia systems for non-technical majors. In *ASEE Annual Conference*

Proceedings, Washington, DC: American Society for Engineering Education, Montreal, Canada, June 2002.

[Burleson02b] W. Burleson, W. Cooper, J. Kurose, S. Thampuran, and K. Watts, Empirical study of student interaction with CD-based multimedia courseware. In *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*, pp. 1430-1443. Washington, DC: American Society for Engineering Education, June 2002

[Caffery05] Glenn Caffery, Ken Watts, and W. Richards Adrion, Assessing Supplemental Courseware in an IT Fluency Course. In *Proceedings of the 35th ASEE/IEEE Frontiers in Education Conference*, pp. T4G-22-27, Indianapolis, IN October 19 – 22, 2005

[Cruz94] G. Cruz and R. Hill, Capturing and playing multimedia events with streams. In *Proceeedings of the second ACM international conference on Multimedia (MULTIMEDIA '94),* pp. 193-200, ACM Press, 1994

[Danneberg97] Dannenberg, Roger B., Are Just-In-Time Lectures Effective At Teaching? June 1997 http://www.jitl.cs.cmu.edu/effectiv.htm last viewed 02/06/06

[eClass] eClass web page, http://www-static.cc.gatech.edu/fce/eclass/ last viewed 5/31/06

[Edelson99] Daniel C. Edelson, Douglas N. Gordin and Roy D. Pea, Addressing the Challenges of Inquiry-Based Learning Through Technology and Curriculum Design. *Journal of the Learning Sciences*, Vol. 8, No. 3&4, pp. 391-450, Lawrence Erlbaum Associates, Inc, Mahwah New Jersey, 1999

[Fletcher00] J. D. Fletcher, J.D. and Philip Dodds, All About ADL, In *Learning Circuits, American Society for Training & Development (ASTD),* May 2000

[Flachsbart00] J. Flachsbart, D. Franklin, and K. Hammond, Improving human computer interaction in a classroom environment using computer vision. In *Proceedings of the 5th international Conference on intelligent User interfaces (IUI '00)*, New Orleans, Louisiana, United States, pp. 86-93ACM Press, New York, NY, January 09 - 12, 2000

[Franklin98a] D. Franklin and J. Flachsbart. All gadget and no representation makes jack a dull environment. In *AAAI 1998 Spring Symposium on Intelligent Environments,* AAAI TR SS-98-02, 1998

[Franklin98b] David Franklin, Cooperating with people: the intelligent classroom. In Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, p.555-560, Madison, Wisconsin, United States, July 1998

[Gamma95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Indianapolis, IN: Addison-Wesley, 1995.

[Gance02] Stephen Gance, Are constructivism and computer-based learning environments incompatible? In *Journal of the Association for History and Computing*, Vol. V, Number 1,May 2002: Retrieved May 13, 2004 download at http://mcel.pacificu.edu/JAHC/JAHCV1/K-12/gance.html

[Harley02] Diane Harley, Jonathan Henke, Shannon Lawrence, Michael Maher, Marytza Gawlik, and Parisa Muller, "An Analysis of Technology Enhancements in a Large Lecture Course at UC Berkeley: Costs, Culture, and Complexity," Center for Studies in Higher Education, UC Berkeley, July 2002

[IBM-BE] IBM BlueEyes http://www.almaden.ibm.com/cs/BlueEyes/index.html last viewed 5/31/06

[IBMTK] IBM Toolkit for MPEG-4 http://www.alphaworks.ibm.com/tech/tk4mpeg4. Last viewed 5/25/06.

[JITL] Just-In-Time Lectures, download at http://www.jitl.cs.cmu.edu/

[JMF] Java Media Framework http://java.sun.com/products/java-media/jmf/. Viewed 5/25/06.

[Kumela04] Dula Kumela, Ken Watts and W. Richards Adrion, "Supporting Constructivist Learning in a Multimedia Presentation System," **Frontiers in Education Conference**, Savannah, GA October 2004

[Klevans97] R. L. Klevans, The Web Lecture System (WLS). *WebNet'97, World Conference of the WWW, Internet and Intranet***,** Toronto, Canada, pp. 669-670, 1997

[LaRose97] R. LaRose and J. Gregg, An evaluation of a Web-based distributed learning environment for higher education. World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications (ED-MEDIA/ED-TELECOM'97), Calgary, Canada, pp. 1286-1287 (1997)

[Lipson01] Alberta Lipson, A Three-School Comparative Analysis of Student Usage Patterns and Attitudes Toward PIVoT," July 2001. download at http://caes.mit.edu/research/pivot/index.html

[Liu01] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz, Automating camera management for lecture room environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '01), pp. 442-449, Seattle, Washington, ACM Press, New York, NY, 2001

[Machnicki02] Erik Machnicki and Lawrence A. Rowe, Virtual director: automating a webcast. Proceedings of SPIE, Vol. 4673, Multimedia Computing and Networking 2002, Martin G. Kienzle, Prashant J. Shenoy, Editors, pp. 208-225, December 2001

[Maly01] K. Maly, H. Abdel-Wahab, C. Wild, C. M. Overstreet, A. Gupta, A. Abdel Hamid, S. Ghanem, A. González, and X. Zhu, IRI-H, A Java-Based Distance Education System: Its Architecture and Performance, *ACM Journal of Educational Resources in Computing*, Vol. 1, No. 1, Spring 2001

[Mediasite] Mediasite http://www.mediasite.com/ last viewed 5/31/06

[Mukhop99] Sugata Mukhopadhyay and Brian C. Smith, Passive Capture and Structuring of Lectures. In *ACM Multimedia 1999,* Orlando, FL, Oct. 30 - Nov. 4, 1999.

[Murhpy02] Laurie Murphy, Kenneth Blaha, Tammy VanDeGrift, Seven Wolfman, and Carol Zander, Active and cooperative learning techniques for the computer science classroom. In *The Journal of Computing in Small Colleges*, Vol. 18, No. 2, December 2002

[Padhye98] J. Padhye, J. and J. Kurose, An Empirical Study of Client Interactions with a Continuous- Media Courseware Server. In *Proceedings of NOSSDAV '98*, Cambridge, UK, July 1998

[Pinhanez95] C. Pinhanez and A. Bobick, Intelligent studios: using computer vision to control TV cameras, *Proc. IJCA'95 Workshop on Entertainment and AI/Alife*, Montreal, Canada, August, 1995.

[PITAC] Using Information Technology To Transform The Way We Learn," President's Information Technology Advisory Committee Panel on Transforming Learning. download at http://www.itrd.gov/pubs/pitac/pitactl-9feb01.pdf

[QTJava] Quick Time Java http://developer.apple.com/quicktime/qtjava/. Last viewed 5/25/06.

[Ray04] Esha Ray, Ken Watts and W. Richards Adrion, Extending Record and Playback Technologies to Support Cooperative Learning. In *Proceedings of the Frontiers in Education Conference (FIE2004),* Savannah, GA, October 2004

[Rui01] Y. Rui, L. He, A. Gupta, and Q. Liu, Building an intelligent camera management system. In *Proceedings of the Ninth ACM international Conference on Multimedia* (MULTIMEDIA '01), Ottawa, Canada, vol. 9, pp. 2-11, ACM Press, New York, NY September 30 - October 05, 2001

[Rui04] Yong Rui , Anoop Gupta , Jonathan Grudin and Liwei He, Automating lecture capture and broadcast: technology and videography. *Multimedia Systems*, Vol.10, No. 1, pp. 3 - 15, Springer Berlin/Heidelberg, June 2004

[Rui03] Y. Rui, A. Gupta, and J. Grudin, and L. He, Videography for telepresentation. In *Proceedings of ACM CHI 2003,* Ft Lauderdale, FL, pp 457–464, 2003

[Schapira01] A. Schapira, K. D. Vries, and C. Pedregal-Martin, Manic: An open-source system to create and deliver courses over the internet. In *Proceedings of the 2001 Symposium on Applications and the Internet*. 2001

[Severance00] Charles Severance, Sync-O-Matic 2000, Michigan State University, East Lansing, http://www.syncomat.com/ last viewed 5/30/06

Shih01] Mei-Yau Shih, Pedagogical Applications of Technology: Tools for Teaching Large-enrollment Classes. In *Proceedings of the AECT 2001 International Conference*, Atlanta, GA, November 2001

[Smeaton97] A. F. Smeaton and F. Crimmons, Virtual lectures for undergraduate teaching: Delivery using real audio and the WWW. In *World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications (ED-MEDIA/ED-TELECOM'97),* Calgary, Canada, pp. 990-995, June 1997

[Steinm01] Arnd Steinmetz, e-Seminar lecture recording and distribution system: a bottom up approach from video to knowledge streaming. *In Multimedia Computing and Networking 2001,* Proceedings of SPIE. Vol. 4312, 2001

[Stern97a] Mia K. Stern, Beverly Park Woolf, and James F. Kurose, Intelligence on the Web? In the *Proceedings of the 8th World Conference of the AIED Society*, Kobe, Japan, 18-22 August, 1997.

[Stern97b] Mia K Stern, Jesse Steinberg, Hu Imm Lee, Jitendra Padhye, and James F. Kurose, MANIC: Multimedia Asynchronous Networked Individualized Courseware, In *Proceedings of Educational Multimedia and Hypermedia*, 1997.

[Stern98] Mia K. Stern,and Beverly Park Woolf, Curriculum Sequencing in a Web-Based Tutor. In *Proceedings of Intelligent Tutoring Systems*, 1998.

[Stern00] Mia K. Stern and Beverly Park Woolf, Adaptive Content in an Online Lecture System. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Trento, Italy, August 2000

[Thampuran01] S. Thampuran, W. Burleson, and K.Watts, Multimedia distance learning without the wait. In *Proceedings of the Frontiers in Education Conference*, 2001

[Thampuran02] R. S. Thampuran, A Multimedia Course Delivery System Combining Web and CD/DVD-Based Technologies. Thesis publication. 2002

[Walker07] Walker, Henry M. "Collaborative learning: a case study for CS1 at Grinnell College and UT-Austin," Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education, San Jose, California, 1997

[Wallace05] Byron Wallace, W. Richards Adrion, Wayne Burleson, Wendy Cooper, James Cori, and Ken Watts, Using Multimedia to Support Research, Education and Outreach in an NSF Engineering Research Center. In Proceedings of the 35th ASEE/IEEE Frontiers in Education Conference, pp. F1E-8-13, Indianapolis, IN October 19 – 22, 2005

[Wallace06] Byron Wallace, Wayne Burleson, Brian Donovan, Jim Kurose, Irene Ros and Michael Zink, Integrating CASA ERC Wireless Networking Research into Education. To appear in *Proceedings of the International Conference on Engineering Education*, 2006

[Zhu04] Zhigang Zhu, Chad McKittrick, and Weihong Li, Virtualized Classroom - Automated Production, Media Integration and User-Customized Presentation. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04),* Vol. 9 p. 138, 2004