Teaching With an Intelligent Electronic Chalkboard

Gerald Friedland
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin (Germany)
fland@inf.fu-berlin.de

Lars Knipping
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin (Germany)
knipping@inf.fu-berlin.de

Raúl Rojas
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin (Germany)
rojas@inf.fu-berlin.de

Ernesto Tapia
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin (Germany)
tapia@inf.fu-berlin.de

ABSTRACT

This paper presents E-Chalk, a software system which transforms a large touch sensitive screen into a smart teaching tool. The instructor writes on the screen using a special stylus and the software emulates a classical chalkboard. The lecturer can paste images to the board, can send queries to remote web services, can activate a computer algebra system, and can paste interactive Java Applets on the board. A copy of the lecture's audio, the board strokes (and an optional video) is stored on a server. The lecture is also transmitted live over the Internet and can be synchronized with teleconferencing systems for student feedback.

The E-Chalk architecture is based on the metaphor of the classical chalkboard, enhanced by intelligent assistants running in the background. One assistant takes care of interpreting the handwritten input of the user. Another is a mathematical formula recognizer which processes handwritten queries for the algebraic server. A circuit simulator recognizes sketches of digital circuits and runs a simulation. An algorithm simulator accepts sketches of graphs as input data and runs graph algorithms, animating them on the screen. Further assistants can be incorporated using the E-Chalk API. E-Chalk is being used in our electronic classroom containing a 6 meter long by 1.15 meter wide rear projection "data wall".

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Distance Learning; H.5.2 [**User Interfaces**]: Input Devices and Strategies

General Terms

Design, Experimentation, Performance

ETP'04, October 15, 2004, New York, New York, USA.

Keywords

Multimedia Educational System, Distance Learning, Digital Ink, Intelligent Agents, Handwriting Recognition, Presentation, Telepresence

1. ENHANCED CLASSROOM TEACHING

When we started the E-Chalk project, our motivation was to replace the traditional chalkboard with a modern alternative based on large computer screens [14]. For the last few years, pen computing in the small (that is, PDAs) has been evolving in parallel with pen computing in the large (contact sensitive whiteboards, plasma screens, rear projection systems). We foresee a moment, in the not so distant future, when large display systems will be available at a fraction of their current cost and will replace traditional blackboards at universities and schools. The question is: What can we offer with this new technology and what kind of teaching will be possible once the "chalkscreen" starts replacing the chalkboard? One important issue is the design of the human-computer interface for such large computer screens, which must obviously be different from the interface for pen computing in the small, and pen computing in laptops.

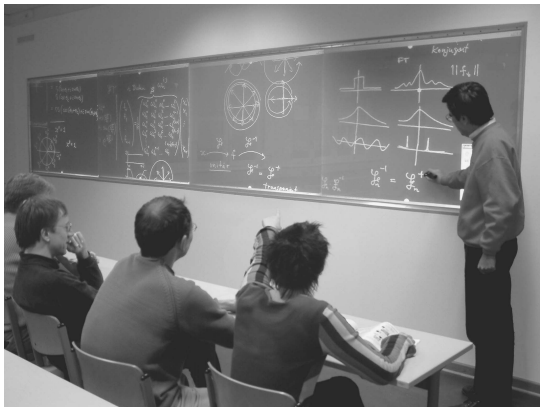

Figure 1: A picture of the electronic classroom with a data wall comprising four rear projectors. The instructor writes with a portable light pen. The screen is 6 meters long and 1.15 meters wide.

We imagine an instructor, present in the classroom, delivering his or her lecture to interested students also physically present. The traditional chalkboard has several advantages for this kind of interactive teaching which has allowed it to survive despite the onslaught of PowerPoint based presentations, very popular at the moment in many universities. One principal advantage of a chalkboard is that it slows the instructor, since diagrams or formulas have to be drawn. Teaching with a chalkboard is like thinking aloud, making clear for the students all the different steps involved in a proof or in the construction of a diagram. This "feature" of the classical chalkboard has proved essential in fields such as mathematics, physics, chemistry, and some others, in which PowerPoint or slide-based lectures are not used by the vast majority of instructors.

Our scenario is therefore a professor, present in the classroom, interacting with the students and adjusting the speed of the lecture to their questions and comprehension. The lecturer is provided with the most advanced technology: a large computer screen which allows her to write and draw with different lines and colors. The lecturer is not alone: a group of software agents is running in the background and is ready to provide answers to queries. Our handwriting recognizer agent, for example, is started whenever the instructor selects a color reserved for it. The instructor can write an arithmetical or algebraic expression, and if the solution is needed to proceed with the lecture, the handwriting recognition agent passes the recognized input to an algebra program (such as Mathematica or Maple) which provides the answer. The instructor continues then with the class. For example, if a plot of the sine function is needed, the instructor can ask Mathematica to provide an image of the plot, which will be pasted on the board and can be further annotated.

The presence of these software agents running in the background transforms the blackboard into an intelligent teaching tool – the board stops being just a passive medium. The students can also write on the board during a class. In our multimedia classroom, several digitizing tablets are located in the tables in front of the students. When the lecturer wants a student to participate and write on the board, the student can do this from its seat. In larger auditoriums it would be possible to use wireless tablets and provide the same interaction opportunity to the students.

2. RELATED WORK

Systems similar in spirit to E-Chalk have been developed using the multicast techniques pioneered with the MBONE [6]. In the last years, however, streamed video has gained the overhand for transmitting classes [1]. It seems that MBONE has been losing relevance in this area while RealVideo and Microsoft Media Player have filled the gap.

Researchers active in the field of "intelligent environments" have explored the integration of intelligent agents in office or meeting rooms environments [5]. Some of the AIRE group projects at MIT have direct relevance for our own work. The O2Flow project is based in MBONE multicasting of streams. The eFacilitator is used to capture and retrieve later handwritten notes from group meetings. Alvarado and Davis [2] studied the automatic recognition of geometric forms in sketches and the resolution of ambiguities. Their system can be used to provide graphical elements for animations. Many other authors have studied the automatic recognition of diagrams, for CAD systems or for enhancing pictures.

There has been much work on electronic classrooms and the use of whiteboards for them. Some systems that come to mind are E-Class [1] and Tivoli, a brain storming tool developed at Xerox [12] and which can recognize and group notes. Scripts can be applied to provide some computational features. The E-Class system at Georgia Tech has been in use for some time, its functionality is a subset of E-Chalk. The Classroom Presenter developed at the University of Washington within Microsoft's Conference XP project, comes closest to our own system. The Presenter provides a whiteboard with real time "inking", which can be annotated by local and remote users. Audio and video are handled by Conference XP. Classes are recorded. Our own system, which predates the Presenter, is based on intelligent agents working in the background, is not proprietary, since it is based on Java, and runs under Windows, MAC-OS X, and Linux [3, 4].

Interfaces for large data displays have been studied by Winograd [7] and Funkhouser [11].They have concentrated on collaborative ubiquitous computing environments [16]. Mynatt et al. [13] studied human interaction with a responsive whiteboard. Sketches drawn in specific frames could elicit different reactions, such as adding two numbers, enhancing a drawing, etc. This is called "behaviors". E-Chalk's intelligent assistants are more general and are not limited to specific frames. The handwriting recognizer, for example, can be active anywhere in the board. Special purpose systems for music scoreboard recognition, image retrieval, and other applications have also been developed.

3. INTERNET CLASSROOM

Once the blackboard has become computer-based, it is natural to provide additional functionality for replaying lectures and live transmission. Figure 1 shows a view of our new classroom with E-Chalk running. The lecturer writes on a contact sensitive surface. The line strokes are processed by a computer, which renders them on the screen. An audio signal is available from a microphone and also an optional video signal from a camera.

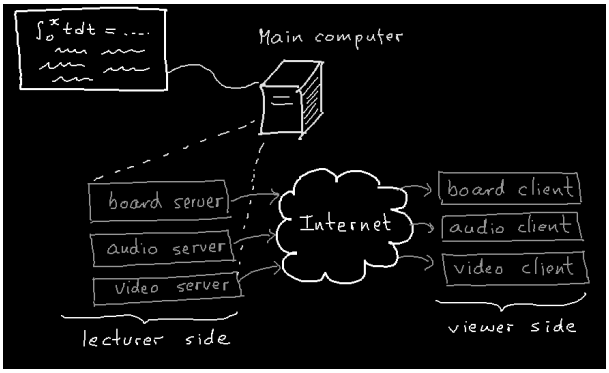

Figure 2: The computer connected to the pen sensitive screen starts three servers, one for each stream. On the receiving side, Applets are started to receive the streams.

Fig. 2 shows the architecture: The main computer starts one software server for each stream. Board events, audio, and video are stored in files in the main computer, but are also transmitted live over the Internet. The instructor proceeds with the lecture and everything happens in the background. A remote viewer, willing to watch a class, needs only to access the web site of the course and click on the link to the live streaming. Since E-Chalk has been written in Java, the remote viewer does not have to install any plug-ins. As long as the browser is Java-capable it can replay any lecture.

On the receiving side, once the connection to the web server is made by entering the URL of the transmission, three Java Applets are started, one for each stream. The user can dispose of them independently, for example of the video Applet if he is only interested

[1] See for example the webcast.berkeley.edu domain.

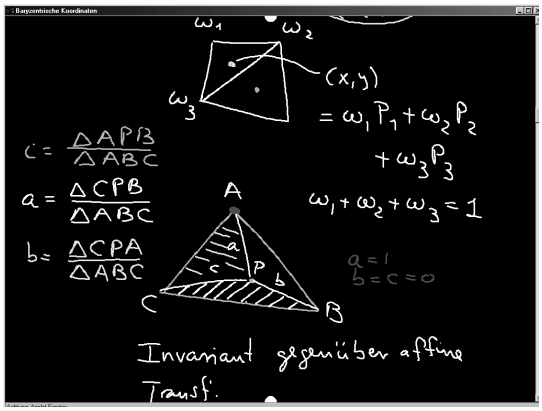

Figure 3: A screenshot of a German lecture about barycentric coordinates.

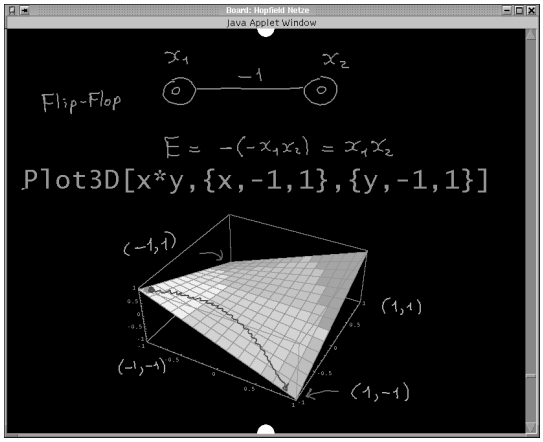

Figure 4: A plot of an energy function provided by Mathematica during a lecture.

in the audio and board events. The E-Chalk streams are synchronized. Buffering is used to mask network jitter during replay. Using Applets as decoder, allows us to improve the system, without explicitly forcing the user to load the latest E-Chalk decoders as frequently happens with Real Media or Media Player plug-ins. The decoders are lightweight Java Applets which are downloaded in a few seconds. The required bandwidth for transmission is low: a few Kilobytes for board events, up to 64 Kbps for audio (the quality of transmission is adjustable), and 64 Kbps for the video stream, sent in a small format.

4. SYSTEM OVERVIEW

E-Chalk is highly configurable. Using a start panel before the lecture starts, the instructor can select images to be used in the lecture (JPEG or GIF) and can bookmark them for future reference. The lecturer can also select the streams to be stored and transmitted (board, audio, video). E-Chalk provides also a transcript in a PDF file of a stored lecture.

Bookmarks can also reference Applets, CGI scripts, and macros. The instructor can bookmark Applets from the Internet or from her hard-disk, and these bookmarks can be used latter during a lecture. Internet scripts and macros are explained further down.

4.1 Pasting Images

E-Chalk is controlled by selecting options from a toolbox, visible on the screen on the sender side. The options are also selectable using a wireless keyboard or buttons on a Bluetooth stylus, to be described in section 6. In order to enrich a lecture, images can be selected from bookmarks and pasted to the board. If there is no bookmark, but the URL of the image is known, the user can enter it directly through the menu. The user can continue annotating the screen. Fig. 3 is an example of a screen produced during an actual lecture. As can be seen, teaching with handwritten input and sketches produces a very peculiar type of lectures. The "ongoing work" character of the class is stressed. The images can even have an esthetic appeal which is lost in other kind of computer generated diagrams.

The main advantage of this kind of user interaction is that classes can be delivered without having to spend hours preparing PowerPoint diagrams. If the lecture is still even more interesting in this way, then E-Chalk saves time and provides richer interaction in class. However, it is true that using a chalkboard is also an art and it takes some experience to make good use of it. Many excellent lecturers always draw their own diagrams by hand, even in slides.

4.2 Computer Algebra Systems

The first intelligent agent that we present in this paper is the algebraic server. Sometimes, during a lecture, a partial result is needed, for example the result of an arithmetical operation or an integral, taken from a table of integrals. Or the instructor would like to annotate the plot of a certain function. An algebraic assistant who can take care of such chores would be very useful. In E-Chalk we have integrated the programs Mathematica (from Wolfram Research) and Maple as such algebraic assistants. Whenever the lecturer needs the result of a computation, they can be called from a menu and the query is typed using a wireless keyboard. Mathematica or Maple answers with a string or an image, according to the query. The Java interface for Mathematica is provided by Wolfram Research, the Java interface for Maple was developed in our group.

Fig. 4 is an example of Mathematica helping during a real lecture. The screenshot shows a portion of class about Hopfield neural networks, and the simple case of a flip-flop network. The instructor has asked Mathematica to plot the energy function of the flip-flop. The image delivered by Mathematica has been further annotated and the class continues.

4.3 Handwriting Recognition

The second intelligent agent to be discussed is the handwriting recognizer. When strokes are drawn in E-Chalk with a reserved handwriting recognition color, the handwritten input is passed to a recognizer. It transforms the input into Unicode character data and can then give it to another application, for example Mathematica. In fact, we would like to avoid having to use the wireless keyboard at all, anytime we want to activate another agent. The handwriting recognizer, which can also recognize drawing gestures from the user, can be used as mediator between the user and the applications running in the background.

We have implemented a mathematical formula recognizer. When a formula is drawn, the recognizer first identifies and groups individual strokes. We use several heuristics to group strokes belonging to the same character, such as the degree of horizontal and vertical overlap, as well as crossings. Once the strokes belonging to a character have been selected, the next step is to simplify their shape. Strokes are given by a sequence of points on the screen, sampled at high rate. Strokes are simplified by eliminating intermediate points in sequences along lines. In curves, only some points are kept, those which can best approximate the curve piecewise within a certain tolerance factor.

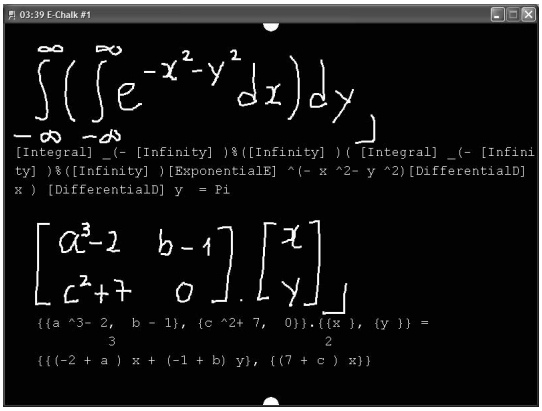

Figure 5: Examples of mathematical formula recognition. The screenshot shows an integral and a matrix vector multiplication. The result of the recognition and the evaluation by Mathematica is shown in typed font under the formulas. The stroke with an angle at the end of the formula is the signal to the algebraic server to start evaluating the expression.

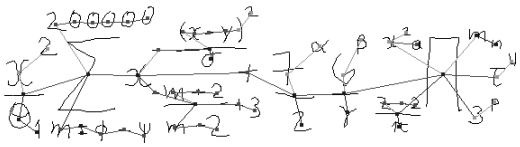

Figure 6: Minimum spanning tree of a mathematical formula, corrected with additional heuristics.

Once a character has been reduced to simplified strokes, represented by just a few points each, we encode each stroke as a vector of features. Each character has been normalized so that it is enclosed by a box of maximum side-length one. The important features for a stroke are the (x,y) coordinates of the start and end points, the coordinates of the centroid of the stroke, the length of the stroke, the distance between start and end point divided by the length of the stroke (that is, how open the stroke is), the accumulated rotation angle when the stroke is drawn, and the coordinates of a few points along the stroke.

Once each stroke has been encoded as a vector, we train classifiers for each character (taken from a training set, produced by several persons). The classifiers are neural networks or support vector machines. There is a classifier for each different number of strokes (since the number of encoding vectors is different). The classifiers recognize each letter. Fig. 5 shows examples of the evaluation of some mathematical formulas recognized by E-Chalk.

Mathematical formula recognition is possible in E-Chalk, because we analyze the input as a two-dimensional structure. Once the individual characters have been recognized, the problem is to assign them a semantic in the formula (as indices, as power symbols, etc.) Our system has been developed thinking of a cooperative user, who tries to write formulas clearly. Otherwise the mathematical formula recognition problem becomes intractable.

In E-Chalk, assigning a role to the individual symbols is based on the computation of a minimum spanning tree for the formula. The most important feature of a formula is its baseline, that is, its central axis. The role of other symbols is determined relative to such a baseline. In the simple formula $x_3^2 = y^5$, the baseline is determined by x, the equality symbol, and y. Powers and subindices can be determined relative to the baseline. Fig. 6 shows a more complex computation of the baseline and spanning tree of a mathematical formula. The branches of the tree are analyzed recursively since they provide the baselines for subformulas of the original formula [17, 18].

4.4 CGI Scripts

There are many Web services available around the clock in the form of CGIs. Many of them receive a string as a query and provide a string or an image as answer. A good example is the LEO English-German dictionary[2], a Web based service which translates words from one language to the other. If a lecturer needs the translation of word “Kreide”, it is possible to send from E-Chalk the string to LEO and receive back the translation “chalk”.

Access to CGI scripts has been implemented in E-Chalk as a very easy way of interfacing third party applications. Anyone with an interesting application, for example a service which provides a digital satellite picture of the current weather over Europe, can set up a Web page which responds to simple ASCII queries. The E-Chalk user only needs to bookmark the Web pages in the E-Chalk start panel. They are called later during a lecture, picking them from the menu of bookmarks. The user can provide additional information and the query is sent to the service.

4.5 Macros

Macros are pre-recorded series of strokes which the lecturer can call and replay on the blackboard anytime during a lecture. Macros were included in E-Chalk as a way of making easier for the instructor to prepare some material in advance, which is used later during a class. One good example are definitions or the formulation of a theorem. It takes some time to write them, but if they have been prerecorded, the lecturer can pull them from a menu and play them on the screen.

E-Chalk provides a recording tool for macros. The user draws on the screen, or on a tablet PC, the portions of the lecture which he or she wants to store in advance. Every macro receives a name, and when E-Chalk is started later in class, the start panel allows the user to bookmark the stored macros and define their replay factor. If the replay factor is high the recorded events will be played fast, if the replay factor is lower than one they will be replayed more slowly than during recording.

Macros are an effective way of scripting a class. Some keywords, definitions, or diagrams can be produced in advance and provide the instructor with a set of “cue cards”, which can be called sequentially to guide the lecture. This “cue card” function is appreciated by lecturers and we have implemented it in E-Chalk.

A macro can contain any recordable E-Chalk event (line strokes, typed text, images pasted on the screen). It is possible, for example, to plot a function with Mathematica and the whole process is recorded. It can be replayed later during the lecture. Even “undos” can be recorded, since they are board events.

Macros can be also generated by programs. A colleague of us has written an animation system in Java which generates the strokes for the animation of an algorithm. Fig. 7 shows two screenshots of the animation. The macro paints and erases lines in order to illustrate the behavior of the Graham scan algorithm. The Java program which generates the macro generates a macro data file containing the appropriate stroke commands.

It is our experience that macros can save time during a lecture, in a sensible way, specially when definitions and complex handdrawn diagrams have to be used. Animations which run within the chalkboard interface and metaphor are very impressive.

[2] http://dict.leo.org

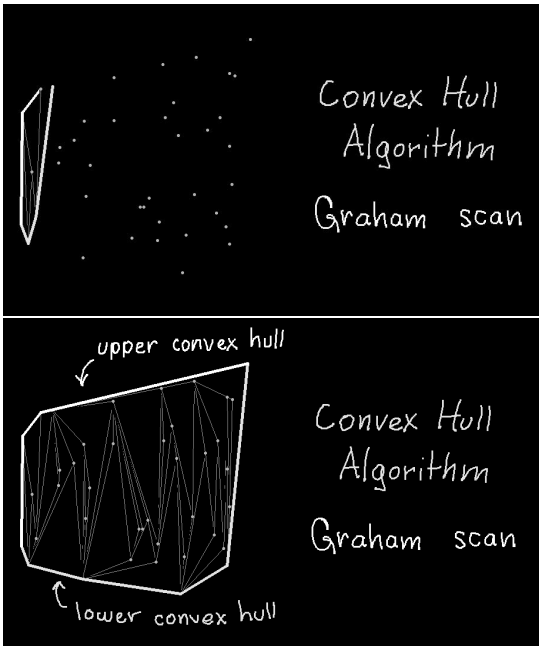

Figure 7: A macro of a run of the Graham scan algorithm. The instructor has written the text, the macro shows how the Graham scans find the convex hull of a random set of points. The two screenshots show two different replay moments.

4.6 Applets

There exist many Java Applets in the Internet which illustrate mathematical ideas and explain physics or even art concepts. Since E-Chalk has been written in Java, we decided to provide a way for the instructors to reuse directly such Applets. An Applet can be accessed from a bookmark, prepared in advance, and can be downloaded from the hard disk or any Web server. The instructor pastes the Applet to the board. A copy of the Applet is sent to remote viewers and is stored together with the recording of the lecture. The instructor can then interact with the Applet, and every Applet event is sent to the remote viewers who see the Applets in action. After a lecture has finished, the remote viewer has an image of the blackboard, as it was developed, and can also interact with the Applet, if desired.

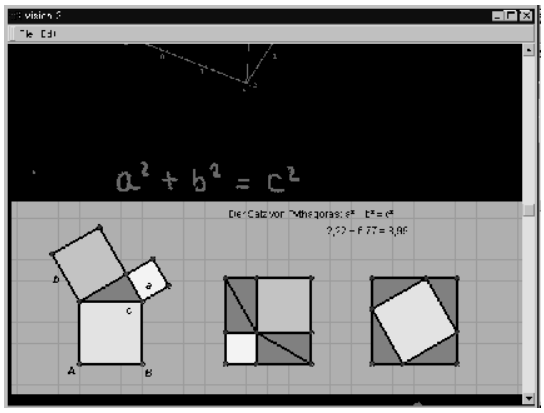

Figure 8: An interactive Applet illustrates Pythagoras Theorem.

Fig. 8 shows an example of an Applet. The Applet is a "visual proof" of Pythagoras Theorem, based on drawing the squares of the sides and of the hypothenuse of a triangle and rearranging them as parts of another square. But it could be that the proof does not hold for other triangles, when we draw them in the same arrangement. The instructor can show that the proof is indeed correct by dragging the corners of the right triangle in the Applet and modifying its size. The visual proof adjusts then automatically and shows its validity.

We have restricted the use of Applets in the version of E-Chalk which we distribute, since Applets from third party programmers are often not cleanly programmed.

4.7 The PDF Transcript

When an E-Chalk lecture is closed, the PDF transcription of the board contents is generated automatically. The transcription can be generated both in color and in black and white. The PDF transcriber tries to arrange the board content in the printed pages in the best possible way, avoiding to break clusters of strokes (diagrams or handwriting).

The transcription is popular with our students, which can concentrate on the class or can make their own annotations besides the board content, knowing that the transcription will be available on the Internet right after the class is finished.

4.8 Exymen: The E-Chalk Editor

Although E-Chalk has been conceived as a tool for capturing live and spontaneous lectures, it is natural that instructors would like to edit the result of a lecture, in order to erase and correct errors, or just to eliminate superfluous portions. An editor for the three data streams is needed.

Exymen [8] has been written as a general editor for streamed data formats. It can handle the E-Chalk format, but it can also handle other formats, just by writing a plug-in for Exymen. Fig. 9 shows a screenshot of the editor. Three streams are visible as jagged lines at the bottom. These are the board, audio and video streams. A cursor allows the user to watch any frame of the synchronized streams. It is possible to select a portion of the three streams and cut it out of the class. It can be reinserted at another position. It is also possible to edit only one of the three streams, for example, the board. If the instructor said the right thing but wrote another, then only the board image needs to be corrected.

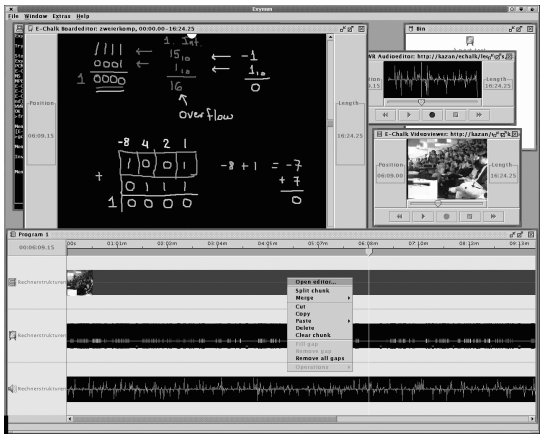

Figure 9: A view of the E-Chalk editor, Exymen, being used to edit three streams. The state of the board at the cursor position is visible (upper left), the video frame (middle right), and the audio signal curve.

Exymen can also export E-Chalk streams to other formats. We can export, for example, an E-Chalk lecture to Microsoft's Media Player format and the lecture can be seen in a PC or PDA without a browser. We have also developed a plug-in for Quicktime and in the future we will have a plug-in for exporting lectures to MPEG 4 format.

5. INTERFACING APPLICATIONS

Web scripts provide the opportunity for third party developers to interface their applications to E-Chalk with minimum effort. Some applications need a stronger coupling with E-Chalk in order to provide an effective service. One ad-hoc example is a game such as tic-tac-toe. We would like the user just to draw a tic-tac-toe field and in so doing, invite the computer to play the game. This requires applications written in Java to be able to read strokes from the screen, and also to be able to draw results on the screen. We would like the application to run in the same computer as E-Chalk and not always from a remote server.

With this need in mind, we designed an Application Interface (API) for E-Chalk. The API allows a program to receive strokes from E-Chalk as Java objects. The program can then process the strokes and can decide what to do with them. The application can answer with strokes of its own, which are then rendered by E-Chalk.

Programs connected to E-Chalk through the API, are stored in the same computer as E-Chalk. A program can be started from a menu of applications. The program requests a certain maximum window size which the user can drag to its size when the application is started. The window is just a region of real state on the otherwise homogeneous chalkboard. Afterwards, all strokes drawn in this region are rendered by E-Chalk but are also passed to the application. This can answer with strokes or continue listening to the stream of strokes. It would be possible, for example, to draw the button controls of a video recorder on the board, and then control an animation to go forward or backward when the buttons are touched.

Fig. 10 shows an implementation of the tic-tac-toe example mentioned above. The application has drawn the field and started playing. The input of the user are the green circles on the field. The game is played until the computer or the human wins or all fields are set. The application can also be stopped by scrolling the play region out of view.

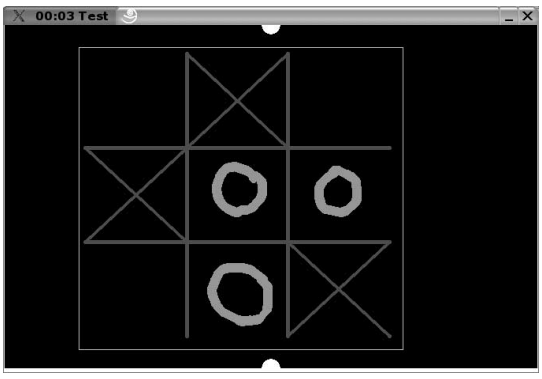

Figure 10: A game of tic-tac-toe. The user plays with circles, the computer plays with crosses.

Fig. 11 shows a more sophisticated example, a digital circuits simulation tool. The user draws the symbols for the logic gates and connects them with "wires" in the E-Chalk board. The program recognizes the type of gates and the connections. Then the user uses a red line to signal a one as input, and a yellow line to signal a zero. The simulation responds by assigning the appropriate colors to all lines. The user can erase an input and change its logical value. The circuit is then recomputed. The simulator also handles timers elements, state diagrams, storing, and loading of circuits for black box reuse.

Another similar tool for simulating pulse coded neural networks is also in development in our group. It will allow neurobiologists to simulate "integrate and fire" neural circuits.

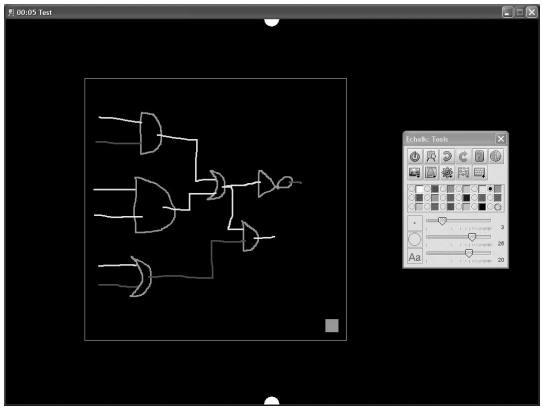

Figure 11: Simulation of a digital circuit drawn by the user. The input is assigned using red for one, and yellow for zero.

Our final example of an application using the E-Chalk API is shown in Fig. 12. The screenshot shows the Prim algorithm for the computation of the minimum spanning tree of a graph running with input supplied by the user through a sketch of the graph. When the program is started, the user can draw a graph. The nodes are represented by circles, edges by lines. The length of the line is the weight of the edge. When the sketch is finished, the algorithm starts and colors each node and edge in the minimum spanning tree being built with another color. The user can see the algorithm running, and can start a new example using a new graph. In this way, algorithms can be explained and tested immediately, with user supplied input, numerical or graphical.

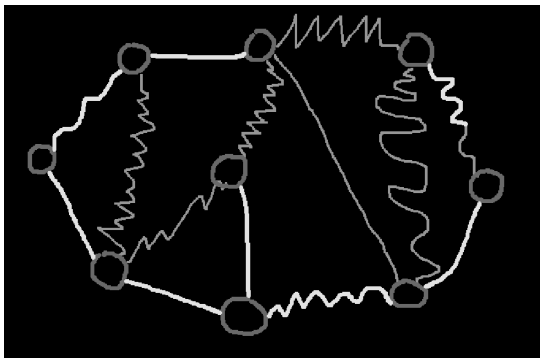

Figure 12: Screenshot of an animation of the Prim algorithm for minimum spanning trees. The user has entered the graph as a sketch. The algorithm is animated by changing the color of the visited and selected nodes and edges.

In the future, it will be possible to couple the handwriting recognizer with this algorithmic animation. We have written a Python

interpreter for E-Chalk. The user can write a program and start it from the board. Corrections to the program can be done just by erasing parts of the code. The next step will be to run algorithmic animations from this code directly, with user supplied input, numerical or graphical.

6. DEPLOYMENT AND EVALUATION

E-Chalk has been in active use at the Free University of Berlin since 2001. Several courses have been offered using the different versions of the system.

One interesting example of an institution using E-Chalk is the case of our colleagues at the Mathematics and Physics Departments of the Technical University of Berlin. All introductory Mathematics courses for engineers are offered in eight large auditoriums and classrooms equipped for E-Chalk. The instructor writes on a contact sensitive whiteboard, and the students can see an enlarged image from a second projector. This provides a very good view for up to 400 students where a traditional chalkboard would be unusable.

Another deployment scenario for E-Chalk which we have also used, is to write the lecture on a Tablet PC, while talking to the students from a podium. The lecturer faces the students and writes, while the students can see an enlarged projection above his head.

At our university we have been using E-Chalk in two different multimedia classrooms. Fig. 1, at the beginning of the paper, shows a view of our newest addition. The large screen is illuminated by four projectors, connected to a single computer with four XGA screens. The stylus used by the lecturer is actually a light pen. When the light pen touches the screen, it is activated. The lecturer does not see the light, but it is visible from behind the plastic screen. Four video cameras located on the backside of the projection screen detect and track the light point. The cameras are connected to a server which can handle four video streams. This machine sends the coordinates of the light pen to the main computer, which paints a point at the location of the light pen.

Our newest light pen has 16 small buttons. A small microcontroller on the pen can detect when the buttons are pressed. The microcontroller communicates via Bluetooth with the main computer and sends the button events. So it is possible, by assigning the buttons to menu options, to control E-Chalk from the pen itself. Changing the pen color can be done by pressing a button. The erase modus can be selected in a similar manner. Undo, redo, and the list of bookmarks can be also activated. This capability is very important in the new classroom with its long screen. The menu has to be at the finger tips of the lecturer. The menu can also be opened at any position on the large screen, but we think that the “hot buttons” on the stylus provide a faster and more convenient way of interacting with E-Chalk.

E-Chalk can be also interfaced with conventional video conferencing software. Fig. 13 shows an example. It is a view from Berlin from a class sent by one of the authors from Stanford University to Berlin. The students see and hear the professor through a Polycom videoconferencing system. The board is transmitted via E-Chalk. E-Chalk records the board events and the audio stream, as usual.

At the end of 2003, E-Chalk was being used at 50 sites, mainly universities in Germany, but also a few in the United States. 700 registered demo versions of E-Chalk have been downloaded since 2001. The Berlin education office is evaluating whiteboards powered by E-Chalk with the intention of installing up to 1000 of them in elementary and middle schools in the next months.

We have made several evaluations of E-Chalk, asking students to fill questionaires [9]. The most extensive was done at the Technical University of Berlin with about 600 questionnaires returned and evaluated [15]. The evaluations have shown that the acceptance by

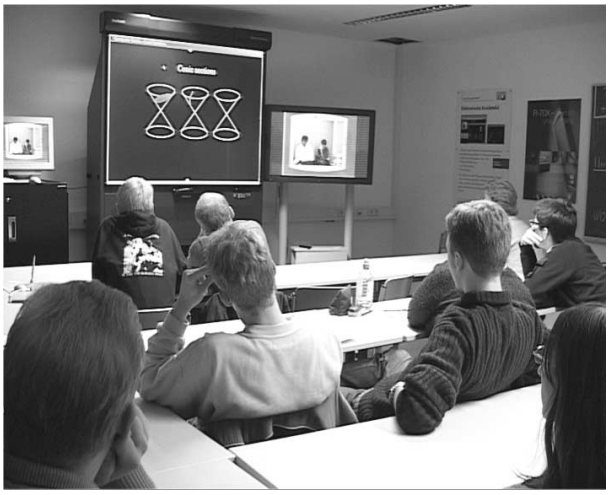

Figure 13: A class sent from Stanford University to the FU Berlin. The students in Berlin see and hear the lecture through a videoconference unit. The board is transmitted by E-Chalk. E-Chalk stores the lecture streams.

the students is very good. However, remote students rarely connect live to a class, they prefer to view lectures already stored in the server.

E-Chalk has been developed with the students present in the classroom in mind, to enhance the classroom experience, and to provide the teacher with several tireless assistants working behind the scenes to improve the class [10]. Therefore, we think that E-Chalk is just a glimpse of what will be possible in the future.

In just a few years new organic screens will make possible to install large flat screens in classrooms. The luminance and contrast should allow to use them under ambient lightning. The user interface will be different to the one we are used to, it will be based on handwriting and gestures of the user. E-Chalk is a first step towards a user interface for such large screens in the context of teaching, but we think that it is the right first step.

7. ACKNOWLEDGEMENTS

The E-Chalk system has been an ongoing project at our lab during the last three years. Several individuals have made important contributions to the E-Chalk architecture or have written applications. U. Raffel wrote the first version of the board software. M. Liwicki wrote the digital signal simulator, and the small tic-tac-toe application. M. Esponda wrote the macro animations of several sorting and graph algorithms. M. Diener wrote the tracking software for the light pen used in our data wall. C. Jantz wrote the Maple interface and a lecturer tracking module. C. Zick built the data wall in our electronic classroom.

8. REFERENCES

[1] G. D. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal, Special Issue on Pervasive Computing*, 38(4):508–530, October 1999.

[2] C. Alvarado and R. Davis. Resolving ambiguities to create a natural computer-based sketching environment. In *Proceedings of the International joint Conference on Artificial Intelligence (IJCAI)*, Seattle, August 2001.

[3] R. Anderson, R. Anderson, C. Hoyer, and S. Wolfman. A

study of digital ink in lecture presentation. In *Proceedings of CHI 2004*, 2004.

[4] R. Anderson, R. Anderson, B. Simon, S. Wolfman, T. VanDeGrift, and K. Yasuhara. Experiences with a tablet pc based lecture presentation system in computer science courses. In *Proceedings of SIGCSE 2004*, 2004.

[5] M. H. Coen. Design principles for intelligent environments. In *Proceedings of 15th National Conference on Artificial Intelligence (AAAI'98)*, pages 547–554, Madison, WI (USA), 1998.

[6] H. Eriksson. MBone: The multicast backbone. *Communications of the ACM*, 37(8):54–60, August 1994.

[7] A. Fox, B. Johanson, P. Hanrahan, and T. Winograd. Integrating information appliances into an interactive workspace. *IEEE CG&A*, May/June 2000.

[8] G. Friedland. Towards a generic cross platform media editor: An editing tool for e-chalk (abstract). In *Proceedings of the fourth Informatiktage 2002, Bad Schussenried.* Gesellschaft für Informatik e.V., November 2002.

[9] G. Friedland, L. Knipping, and R. Rojas. Mapping the classroom into the web: Case studies from several institutions. In C. T. András Szüks, Erwin Wagner, editor, *The Quality Dialogue: Integrating Cultures in Flexible, Distance and eLearning*, pages 480–485, Rhodes, Greece, June 2003. 12th EDEN Annual Conference, European Distance Education Network.

[10] G. Friedland, L. Knipping, R. Rojas, and E. Tapia. Web based education as a result of ai supported classroom teaching. In *Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003 Oxford, UK, September 3-5, 2003 Proceedings, Part II*, volume 2774 of *Lecture Notes of Computer Sciences*, pages 290–296. Springer Verlag, Heidelberg, September 2003.

[11] T. Funkhouser and K. Li. Computer graphics and applications. onto the wall Large Displays. *Special issue of IEEE*, 20(4), August 2000.

[12] T. P. Moran, W. van Melle, and W. Tivoli. Integrating structured domain objects into a freeform whiteboard environment (demonstration). In *Proceedings of the International Conference on Human Factors in Computing Systems (SIGCHI)*, Hague, Netherlands, April 2000.

[13] E. D. Mynatt, T. Igarashi, W. K. Edwards, and A. LaMarca. Flatland: New dimensions in office whiteboards. In *Proceedings of the International Conference on Human Factors in Computing Systems (SIGCHI)*, Pittsburgh, May 1999.

[14] R. Rojas, L. Knipping, W.-U. Raffel, and G. Friedland. Elektronische Kreide: Eine Java-Multimedia-Tafel für den Präsenz- und Fernunterricht. In *Tagungsband der Learntec*, volume 2, pages 533–539, October 2001.

[15] J. Schulte. Evaluation des Einsatzes der Software E-Kreide in der universit ären Lehre. Master's thesis, Technische Universität Berlin, Institut für Sprache und Kommunikation, November 2003.

[16] A. F. T. Winograd, B. Johanson. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2):67–75, April/June 2002.

[17] E. Tapia and R. Rojas. Recognition of on-line handwritten mathematical expressions using a minimum spanning tree contruction and symbol dominance. In *Fifth IAPR International Workshop on Graphics Recognition (GREC)*, July 2003.

[18] E. Tapia and R. Rojas. Recognition of on-line handwritten mathematical formulas in the e-chalk system. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, August 2003.