

# Chapter 1

## Introduction and Related Work

*Computers promise the fountains of utopia, but only deliver a flood of information*

Langdon Winner, Mythinformation, Whole Earth Review, Jan. 1985

### 1.1 The E-Learning Landscape

The history of-computer supported education is rooted in the early era of mainframes. In 1963, the *PLATO system* [Woo94] (Programmed Logic for Automated Teaching Operations) was developed at University of Illinois. It had custom-built multimedia teaching terminal stations connected to the mainframe. A proprietary language called *TUTOR* was created for authoring educational software. More than 15,000 hours worth of instruction material were developed for *PLATO*. Features like online chat and bulletin-board notes were added in the early 1970s, long before the Internet. In 1976 Control Data Corporation (CDC) established *PLATO-IV* as a commercial educational product, with its successor still around [75].

Recent years have witnessed considerable activity in the field of computer-aided education. With an access to the Internet being almost omnipresent, people envision a new age of learning: the learners will be free to learn when and where they want, learning efficiencies will be heightened by a multimedia-enriched learning experiences, and training costs can be cut by reusing teaching content.

#### Production Costs

Unfortunately, creating e-learning material is a laborious process. Production costs are reckoned to be in the range of 50 to 200 man hours for one hour of learning content. [CMMS03] reports on the development of static or dynamic HTML pages:

*On the average, one can calculate one person working for a half to up to a full year for the preparation of electronic learning content for only one course.*

While the costs of any software and hardware equipment amortize rapidly if enough material is produced, the huge costs in personnel make it economically not viable unless the content is either aimed at a very large audience or can be reused many times. For the teaching at universities, the situation is particularly grave, as the contents taught tend to change very fast.

A cause for this tremendous effort is that traditional teaching know-how does not easily match with contemporary authoring tools. Apart from technical effort it requires a huge amount of work to structure didactic content for the Web, even if presented only linearly. As a side note, the e-learning community often argues for producing highly interactive material supporting constructivistic style learning, following the learning theory rooted in the works of Jean Piaget and Lev Vygotsky. According to constructivism, knowledge cannot be simply transmitted. Instead the learners have to actively construct their understanding of a subject. In reality, most Web-based learning courses rely on delivering information and rote learning. Interaction is usually limited to online multiple-choice tests and possibly a few interactive experiments such as Java Applets or flash animations. Complex interaction and feedback are almost non-existent [MN02], at the very best providing chat, e-mail discussions, or automatically examined multiple-choice tests [Tsi99].<sup>1</sup> Since even simple and linear material is labor-intensive and expensive to produce, more complex approaches are rarely realized and almost always stay in the realm of theoretical concepts.

On the other hand, a huge amount of content is already produced on a regular basis in traditional teaching. As [Tsi99] puts it:

*Universities generate content every day through their courses and seminars. They then throw it away. There is a certain charm with this approach but it is not cost effective.*

Trying to avoid the expenses of standard e-learning module authoring, many universities resort to mere video capturing of their standard lectures, see for example [106].

This approach has the advantage of making use of existing teaching qualifications of the lecturer, instead of requiring the lecturer to acquire new teaching skills. If the recordings manage to transport the feeling of the lecture, they can produce high-quality teachings as a kind of by-product of traditional teaching. The resulting e-learning content will be linear, but then everyday academic teaching by lecture<sup>2</sup> is linear by nature. Also, the instructor does not have to have intimate knowledge of the production process. Normal authoring systems for e-learning modules either require the teacher to learn how to operate the authoring software or the authoring process to be handled by a team of at least

---

<sup>1</sup>In fact, with the popularity of constructivistic teaching, even systems offering the replay of plain lecture recordings have claimed to be interactive and constructive, just because of the control of the recordings replay. The developers of *MANIC* (described in Section 1.6.1) write in [SSL<sup>+</sup>97]: *To provide for interactivity, students are given the opportunity to browse the material at their own pace, stopping and starting the audio at will* and in [SVPM01] they even claim: *We have based the overall design of the system in Constructivism [...]. We believe that the features of the MANIC system described above (audio controls, controls to move back and forth between the slides, a table of contents, and a search engine) allow for a greater participation of the students in their learning process.*

<sup>2</sup>The instructivistic teaching style embodied by lectures can be clearly considered as the dominant teaching form in the natural sciences and in engineering subjects. Even though criticized for its one-way communication, teaching by lecture nevertheless remained popular because of its efficient way to educate large numbers of students.

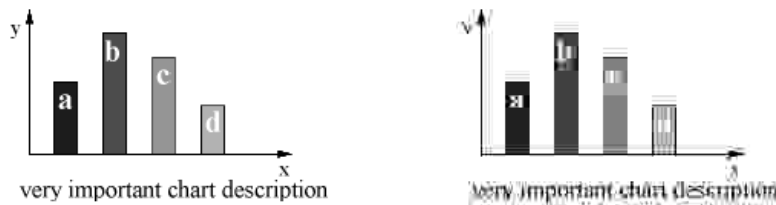


Figure 1.1: Effects of frequency-based lossy compression on sharp edges. Left: image encoded with lossless compression (as PNG). Right: same image compressed with reduction of higher frequencies (JPEG).

two specialists: one who knows how to operate the technology and one who knows the subject and the didactics.

Still this approach needs extra personnel present during the recording to handle the camera and the audio hardware, plus provisions to put the recording online in a digital form. Naive recording setup will cause these systems to produce poor audio and video recordings from live lectures as their codecs assume a clean signal. In practice one needs high-quality sound equipment and qualified technical staff to eliminate audience noise, reverberation effects, changes in illumination, etc. Even when tailoring this approach to a light-weight creation process, Carnegie Mellon's *Just-In-Time Lectures*<sup>3</sup> calculated a post-processing time of about 30 hours per one hour of lecture, not including the time for recording and preparation. For the Microsoft Technical Education Group (MSTE), costs are reported to amount to more than \$500 per talk for their video-taped trainings, with the expenses primarily being costs for staff to record and put the talks online [LRGC01].

Not only that this is still quite expensive, but encoding with off-the-shelf Internet video tools is inadequate for lecturing content. Writing and drawings, from slides or from a blackboard, are not encoded appropriately. Compression of a single video frame with state-of-the-art video encoding technology relies on dropping the higher-frequency parts from images resulting in the loss of sharp edges. Either the content becomes unreadably blurred or, using only weak compression, the video stream takes up lots of bandwidth. JPEG image compression is based on the same approach, see Figure 1.1 for an illustration of the effects. Thus these systems require the instructors to modify their teaching style for the sake of tele-presence quality [Tsi99]:

*[...] any teaching materials must be prepared with consideration of legibility to tele-viewers. The writing on blackboards or the transparencies shown on overhead projectors may be clearly visible to classroom participants but extremely hard to read for remote participants.*

Even then, considering the bandwidth restrictions the typical remote user faces at home, low resolution and lossy encoding often result in unacceptable quality.

One approach to avoid badly-encoded writing is to send lossless encoded presentation slides of the talk separately instead of encoding them as video

<sup>3</sup>The *Just-In-Time Lectures* concept is described in Section 1.8.2.

image. They are either sent as a collection of all slides, requiring the remote user to trigger the slide transition manually<sup>4</sup>, or their display is synchronized with the audio/video stream<sup>5</sup>. Some specialized lecture recording tools allow to give presentations and automatically store the transitions for replay, others require to add the transition times by extra manual effort in a post-processing phase.

### Teaching with Slideware or Desktop Environments

Teaching with slideware tools like PowerPoint is prevalent, regardless of the lectures being recorded or not. However, this has been heavily criticized [Cre97]. The advantage of slides is their easy reuse and the wide availability of slideware tools on computers. On the other hand, they are devised for presentations and not for teaching. Edward Tufte argues that PowerPoint is a good tool to convincingly “sell” something while complex arguments tend to get lost [Tuf03a, Tuf03b].

As stated by Richard Clark, the human brain can be easily overloaded by the sensory input that e-learning and multimedia technology is capable of generating [Cla99]. While obviously there are people who can give great classes using slide presentations, these tools foster a tendency to overwhelm learners with rapid deliverance of information. The lecturers, of course, already have a deeper understanding of the subject and they often tend to continue with a speed too fast for their students to follow when not restrained by the teaching technique. Also, classes given with slideware tend to be much more predetermined and less spontaneous. To use the words of a university lecturer, “*PowerPoint sucks the life out of a class*” [And04].

Some lecture-recording tools record work by recording the computer desktop<sup>6</sup>, see Section 1.8.1. While technically this allows an instructor greater freedom and encompasses the recording of slide shows, it is an unsuitable approach for teaching. The desktop metaphor is tailored for personal work, as a virtual extension of the physical desktop, meant to be used with a mouse and a keyboard by a single person and not to be shared with an audience. The desktop metaphor needs sustained attention to be operated due to the prevalence of modes.<sup>7</sup> Disruption of the users workflow is common, for example by requiring complex interactions with dialogs. The universality of the desktop paradigm turns into a drawback in the lecturing situation. As a teacher giving a lecture, one is already quite busy with getting the content across, even without having to deal with desktop dialogs, like creating folders or killing the MS Office Assistant, that are a source of distraction for both lecturer and students.<sup>8</sup>

---

<sup>4</sup>See for example iLectures [36].

<sup>5</sup>See for example the *Cornell* and the *BMRC Lecture Browser* [5] described in Section 1.6.2.

<sup>6</sup>Again, most of these systems have the encoding problem mentioned above, as almost all use standard video compression codecs.

<sup>7</sup>See Section 2.2 for a short discussion of modes in user interfaces.

<sup>8</sup>These examples are not as uncommon as they might appear at first glance. Both have been witnessed more than once by the author in sales presentations of whiteboard software for teaching.

## 1.2 E-Chalk: The Idea

The main fault in the approaches described lies in being technology-driven, focusing on what can be done with computer today, instead of being led by the demands of teaching.<sup>9</sup> Looking instead for established teaching techniques, one finds that the chalkboard has been an unmatched teaching tool for ages in many disciplines. In 1855, the abolitionist Samuel Joseph May wrote about the introduction of the blackboard to classrooms<sup>10</sup>, being at his time the most modern instruction technology:

*[...] in the winter of 1813 & '14, during my first College vacations, I attended a mathematical school kept in Boston by the Rev. Francis Xavier Brosius [...] On entering his room, we were struck at the appearance of an ample Black Board suspended on the wall, with lumps of chalk on a ledge below, and cloths hanging at either side. I had never heard of such a thing before. There it was forty-two years ago that I first saw what now I trust is considered indispensable in every school the Black Board and there that I first witnessed the process of analytical and inductive teaching.*

The chalkboard *is* an adequate interface metaphor for a common display used in teaching an audience. It provides a shared view for instructor and students. The board ensures that information stays available, providing context for talk and discussion. The learners can see how ideas are developed rather than being overwhelmed with final results – they are helped to follow the conceptual process. The teacher is slowed down to the speed of his or her handwriting, giving the students time to follow the lecturer’s train of thought.

The “chalk and talk” approach results in a much more flexible teaching style than relying on prepared slides. Working on a chalkboard supports creative thinking, illustration and sharing. Board drawings can also be used to draw attention to details using circles, arrows, underlines, checks, grouping, etc. The inherent impreciseness and vagueness of freehand drawings holds extra information. With these great qualities in teaching, it comes as no surprise that the chalkboard is still so popular in teaching in many disciplines, especially for subjects where complex reasoning has to be taught, as in mathematics and the natural sciences.

These considerations inspired the development of a system called E-Chalk [22]. Ideally, the lecturers are enabled to teach with the system like with a regular blackboard and produce distance teaching material as a by-product. During classroom teaching, the lecturer works directly on a pen-active wall display. The system tries to enhance teaching in the classroom by allowing the instructor to integrate multimedia elements. At the same time, the lecture is being saved and transmitted live over the Internet without extra effort required of the instructor. The system transmits audio, video, and the animated board image of the lecture. A PDF file is also generated as a static copy of the board content for printing. The goal is to preserve the didactic advantages and the easy handling of the traditional chalkboard, while augmenting the classroom

---

<sup>9</sup>This is a problem that is not specific to the e-learning domain. As argued by Donald Norman, the prevalence of PCs result in most information technologies being technology-centered instead of being human-centered [Nor98].

<sup>10</sup>Quote cited according to [And04].

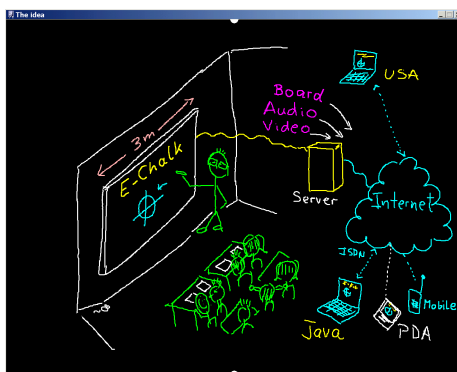


Figure 1.2: The idea of E-Chalk represented as an E-Chalk sketch.

teaching and extending its reach to distance learning. See Figure 1.2 for an overview sketch of the system.

In order to take advantage of the lecturers already being proficient in chalkboard lecturing, the tool to be developed should closely resemble the chalkboard in handling and avoid burdening the user with technical details as far as possible. Integrating the system in a non-interfering way into the users' everyday environment does not only ease the handling of the system, it is also essential for the system's widespread acceptance. As [BAT99] noted, "*our experience [...] is that even for scheduled activities like classes, users are reluctant to spend a few minutes setting up the classroom.*" With this aim of blending E-Chalk into the regular teaching environment, the design follows the philosophy of Ubiquitous Computing [WGB99].

Similar observations hold true for the remote learners. Most systems for lecture-recording require remote learner to install a special client software often designed as a browser plug-in.<sup>11</sup> This introduces a psychological barrier for first-time users, compare for example [Nie99]. Moreover, remote learners often do not have the skills or even the permissions (e. g. on campus computers) to install such a client software. For example, [ZS02] reports students having trouble installing the *Transparent TeleteachingTool* (TTT) player<sup>12</sup>, even though they "only" had to install Java, JMF (Java Media Framework), and the TTT application.

The learners should not be required to install any proprietary viewer software or codec and should not be restricted to a certain operating system. Therefore it was decided to realize the replay as a Java Applet running in any Java-enabled browser.<sup>13</sup>

The system is not designed to replace teaching in the classroom. The recordings should "capture the live experience" of the lecture's natural flow, as well as having the teaching style formed by interactions with a learning audience. The

<sup>11</sup>For example *AOF*, *Lecturnity*, and *Camtasia* require proprietary player software. Others require to install certain Windows Media codecs or rely on the *RealPlayer*, or a combination of these techniques. See Sections 1.5 to 1.8 for details.

<sup>12</sup>See Section 1.8.1 for a description.

<sup>13</sup>While there are some browsers that are not able to run Java Applets, this is a usage requirement that is much lower than all available alternatives that allow for dynamic playback. It was also decided to stick with Java version 1.1 for the replay, as most pre-installed Java browser plug-ins are still of this rather early Java version.



Figure 1.3: The *ChalkBoard PowerPad* by Chalkboard Inc. was an early example of a digitizer tablet. It was available for the C64 and the Atari 800/XL/XE and came with educational software. Additional edutainment programs were sold for the *PowerPad*. Each came with a plastic clamshell box to serve as a customized template that was laid over the *PowerPad*'s surface. Image from [51].

approach merges classroom teaching, distance teaching, and the production of courseware into a single task.

Not only does the replay allow for pure distance-teaching scenarios, it also relieves the classroom students of the burden of writing down everything and gives them more time to participate in the lecture, as the recording is a dynamic script of the class where the teacher's side notes are not lost. Experiences showed that automatically-generated script does not eliminate note-taking by students. Instead, their notes become less literal copies of the board content and more summaries capturing "*the essence of the lectured material in their own words*" [TA99].

### 1.3 Board Hardware

When using the E-Chalk system, the instructor needs to have some kind of electronic pen-input device and a display for the audience. In situations where the hardware needs to be portable, a Tablet PC with a data projector to display the computer screen for the class is a viable option. Alternatively, digitizer tablets can be used. The tablets come in a broad range of variants, including tablets with integrated LCD screen [104], greatly easing the required hand-eye coordination in writing and drawing. Both Tablet PCs and digitizer tablets track the pen using electrical induction.

An early example of digitizer tablets which even used the chalkboard metaphor is shown in 1.3. The first commercially available notebook-sized computer with integrated input pen was the *GRiDPad* from GRiD Systems, released in September 1989. Its operating system was based on MS-DOS. In 1991 another tablet computer, the *Momenta Pentop* [Mom91] from Go Corporation, became available, this time with a dedicated operating system, called *PenPoint*. These early examples were commercial failures, suffering from available handwriting recognition not being sufficient for user requirements, and from the product's high cost and weight. The *Momenta*, for example, weighed seven pounds and had a purchase price of about \$5,000. So pen computers beyond PDA size did



Figure 1.4: E-Chalk with a plasma-screen display, here in combination with a digitizer tablet (Wacom *Intuos* [103]) as input device.

not become popular before the Windows Tablet PCs, released in late 2002.

When hardware mobility is not an important factor, the instructor should ideally be enabled to write directly on the shared screen. This way, learners do not have to look back and forth between the talking teacher and the screen, as both are in the same place. For this, technologies for pen-writing on wall displays are required. Display technologies of such solutions usually use front or rear projection. In early E-Chalk experiments, plasma screens with touch technology were also tested [Raf00,RKRF00,RKFF01], but the susceptibility of these screens to burn-in effects made them an inapt choice. See Figure 1.4 for an example usage.

A front-projected solution is much cheaper than a rear-projection system and easier to move, as rear-projection systems tend to be quite heavy. However, the projection of the latter is superior in brilliance, is less susceptible to interference from sunlight, and the instructor also does not cast shadows when he or she stands in front of the display. With front-projection, the instructor might look into the glaring projector when turning to the audience. Moving the board changes the relation between position on the board and in the displayed image, and requires recalibration of the system. In the near future, organic displays are expected to be available for large screens, eliminating these drawbacks in display technology.

### 1.3.1 Digital Boards

Xerox PARC not only pioneered the GUI with the introduction of the Xerox Star System [JRV<sup>+</sup>89], its former division LiveWorks also produced the first digital whiteboard, the *LiveBoard* [EPT<sup>+</sup>92]. The system used a rear-projection screen controlled by a built-in workstation or PC<sup>14</sup>, and a set of tracked pens for different colors. The system allowed overlay annotations and had the ability to interoperate seamlessly with remote *LiveBoards* in other locations.

The *BrightBoard* [SFR96] and the *ZombieBoard* [Sau98, BBJ<sup>+</sup>98] are both regular whiteboards with markers, where the whiteboard actions are tracked by

<sup>14</sup>The research version used Solaris workstations. The version later sold as a commercial product used PCs running Windows 95.



a video camera. Drawing commands triggered actions by the system, like a save of the video-scanned board content.

A number of digital whiteboards use basically the same tracking technology as digitizer tablets. The boards mainly differ in being larger than the pads and being mounted upright. Examples include products like the *Numonics IPM* series [69], the Promethean *ActivBoard* series [77], the *GTCO whiteboards* [31], the *TeamBoard* [91], and the Panasonic *Panaboard* [73]. They can track the pen both when it is near the board but not touching it (similar to hovering mouse cursor) and when the pen is pressed down for writing (similar to a mouse drag). Many also feature an extra button on the pen, allowing to emulate a second mouse button.

With the products *Mimio* [102] and *EBeam* [21] one can turn regular whiteboards into digitizing ones. The product is compromised of sheaths for normal whiteboard pens and sensors to be attached to the whiteboard by vacuum cups. When the pens are pressed down for writing, they are located on the board by a combination of ultrasonic and infrared signals. In early experiments of the E-Chalk project with *Mimio*, the device exhibited observable delay, too large for it to be conveniently used in combination with a front projection.<sup>15</sup>

The *SmartBoard* products from Smart Technologies [88] include both front- and rear-projected systems. Their pen input is based on touch technologies, registering pressure on a touch-sensitive film by measuring changes in the electrical resistance. A finger can be used instead of a pen, but the input system does not distinguish between a mouse drag and a mouse move. The *SmartBoard* handles several colors of writers and an eraser tool. This is done by trays to keep the writing tools in. The system assumes the writing tool to be used for which the associated tray is empty. When the user holds a pen in each hand or puts a pen in the wrong tray, things can quickly become confusing.<sup>16</sup>

PolyVision Visual Communications [76] *Webster LT Series* uses lasers to track the reflective bands of the pens and erasers. Different pens have different reflective patterns, so that they can be distinguished. The *Webster TS Series* from the same company uses resistive sensors.

The *StarBoard* Series from Hitachi Software [35] features both front- and rear-projected whiteboards. Some of the pen-tracking systems for the front-projection systems are based on electrical induction, others use a combination of infrared and ultrasonic tracking. The rear-projected *StarBoard R-70X* tracks the pen with an infrared laser.

## 1.4 Pen Computing Software

A number of pen-input metaphors have been considered, including writing in notebooks, for example *Filochat* [WHW94] and *Dynamite* [WSS97a], both described in Section 1.5.1, on cocktail napkins [GD96] or onion skins [Kra94], the flip-chart metaphor, e. g. *Flatland* [IMEL99], see Section 1.5.4, or the oil-painters-palette-and-easel metaphor, see *M-Pad* [Rek98] in Section 1.5.7. The

<sup>15</sup>A project spawned from these experiments for developing GNU/Linux driver for several whiteboard interfaces including *Mimio* [19].

<sup>16</sup>Even worse, in one of the schools where the use of digitizing boards was evaluated within the *CidS!* (*Computer in die Schulen!*) project – see Section 8.1, the eraser tool was stolen, causing the board to interpret all writing actions as being done with the eraser. A dummy had to be put into the eraser tray as a workaround.

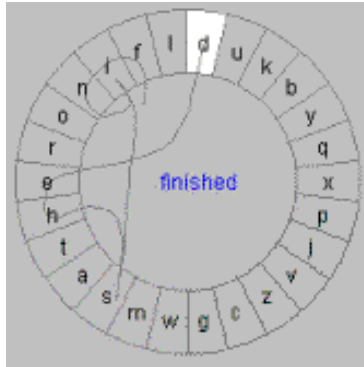


Figure 1.5: Writing the word “finished” with a single stroke using *Cirrin*. Figure from [MA98].

Apple *Newton* MessagePad [App93] used writing on sheets of paper as a metaphor. The interface of the *PenPoint OS* [CS91] was based on a notebook metaphor, and the use of the pen as the primary, and for most uses only, input device. It was a product of the Go Corporation, the earliest operating systems written specifically for graphical tablets and personal digital assistants, running for example on the *Momenta* and AT&T’s *EO Personal Communicator*.

For creating text input with a pen or mouse, a number of approaches have been devised. The simplest approach is the use of a *software keyboard* [SRS<sup>+</sup>93], like the one provided on Windows platforms. Handwriting recognition is another popular approach; i. e. the *CalliGrapher* [74] recognition engine by Paragraph International was integrated in the Apple *Newton*, and the recent version is part of the Windows CE and XP Tablet editions. The Xerox *Unistroke* Recognizer [Gol97, GM93] is a gesture-based system loosely based on the Roman alphabet, using a single stroke per letter. A similar system is the PalmPilot [72] OS’s *Grafitti*.

To submit commands with pens, context menus with a dial instead of a linear arrangement are often used. These *pie menus* [Hop91] allow the user to learn the command selections in a kind of stroke gestures. On the downside, they are hard to extend when new commands are added, and for control by pen input they suffer from occlusion by the writing hand. *T-Cube*<sup>17</sup> [VN94] is a system to type input in the absence of a keyboard (i. e. with a mouse or a pen) using pie menus, predicting this being faster than linear menus or software keyboards, according to Fitts’ Law<sup>18</sup> [Fit54]. *Quickwriting* [Per98] also uses circular menus, but tries to reach interruption-free input sequences by returning to the home position between keys instead of signalling each key input by lifting the pen. *Cirrin* [MA98] is a key-input software originally designed for users with repetitive strain injury. Like *Quickwriting* it allows to enter the keys without having to lift the pen before a word ends, quickly training the user to remember the key input motions as kind of gestures. Its main difference to *Quickwriting*

<sup>17</sup>The input system called *T-Cube* is not related to the distance-teaching system *t-Cube* from Universität Trier described in Section 1.8.8.

<sup>18</sup>Fitts’ Law states that the time to acquire a target is proportional to  $\log(d/s)$  with  $d$  the distance of movement from start to target center, and  $s$  is the size of the target.

is that it arranges the letters in a circle, giving a rather fine-graded “wheel” of choices instead of a hierarchical organization, see Figure 1.5. For standard commands instead of key inputs, the *flowmenus* described in Section 1.5.6 also use the homing strategy to allow for rapid input of consecutive commands.

For entering text with *Dasher* [WBM00], the user selects the rectangle for the next letter, where size and position of the rectangles depend on the probability of the letter as the next input element. This makes it easier and thus faster to select a probable letter. Note that the size-determining probabilities are language-model dependent. The drawback of the system is that it requires constant visual attention to operate, and it is still slower than the traditional keyboard.

## 1.5 Pen-based Office and CSCW Tools

Boards are often used in companies as a tool for presentation and brainstorming support. In offices, they are also used as memory-extension tools as well as a semi-public drawing area shared with collaborating visitors [Myn99]. When digitizing whiteboards entered the market, the first applications were in commercial settings. Other pen-input-enabled electronic devices are also prevalent in commercial contexts, like PDAs and Tablet PCs.

As a consequence, pen-based interactions have mostly been researched in the context of computer-supported collaborative work (CSCW) and office tools. Most CSCW tools provide shared workspaces according to the WYSIWIS principle (what you see is what I see) and often allow one to capture the results of a session (“collaborative capture systems”), though in most cases they allow only a static snapshot of the result.

Many of the research projects are closely related to the idea of Ubiquitous Computing. In fact, according to [Wei93], the first Ubiquitous Computing technology to be deployed was a digital whiteboard, the Xerox *LiveBoard*.

### 1.5.1 Personal Note-Taking

Storing pen-based annotations as personal notes is one of the main applications of PDAs and Tablet PCs, coming with pen-input software like the *aha! Inkwriter* [aha93] and the *Windows Journal*. In contrast to the CSCW tools, these types of notes are not primarily meant to be shared with co-workers.

The objective of the *Dynamite* [WSS97a, WSS<sup>+</sup>97b] (dynamically organized ink and audio notebook) is to maintain to-do lists, personal diaries, address lists, and other notes. It records pen actions, audio may be added optionally. The handwritten notes are given creation timestamps. For retrieval of the notes a keyword-query interface was created.

*Filochat* [WHW94] aims at replacing the traditional dictaphone. It uses an LCD tablet for indexing the recorded audio by hand-written notes. A header with time and date information plus space for a handwritten topic and name is automatically generated for each new notes section. The recorded audio can be accessed for replay by a seek option or by selecting the associated note.

The *Audio Notebook* [Sti96, Sti97, SAS01] is another approach to combine a digital audio recorder and a notebook, but it is built as a physical device using real paper for note-taking, see Figure 1.6. It relies on a digitizing pen for

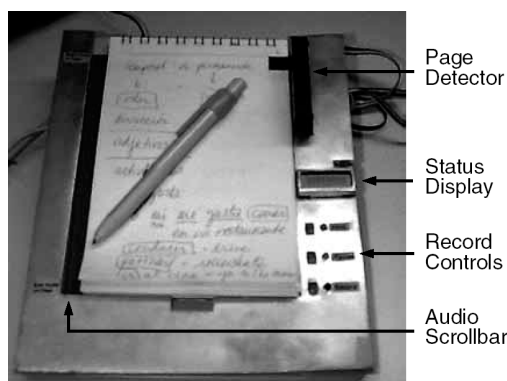


Figure 1.6: An *Audio Notebook* prototype. Figure from [SAS01].

computerized access to the handwriting and the paper sheets used are marked to be recognized by the device.

*Marquee* [WP94] is a system for annotating videos with handwritten notes. For synchronizing the notes with the video, time zones are created using a horizontal-line gesture. Handwritten keywords are manually converted to text for indexing purposes.

### 1.5.2 Commercial Digitizing Whiteboard Software

Most customers of digitizing whiteboards buy the hardware to equip their meeting rooms. As a result, the software for these boards focuses on meetings and presentations. They build on the standard desktop GUI and allow pen-based annotations. Some support desktop sharing for interaction with local co-workers using personal computers, some allow for distributed work. For example, the products of *Centra Symposium* [9] and *WebEx Training Center* [107] aim at presentations to remote audiences.

Often, these tools can save the board content in static pages; *ACTIVstudio* for example, that comes with the Promethean *ActivBoard* [77], allows to save materials as HTML pages or as Microsoft PowerPoint presentations. Some even have limited replay capabilities, like the software of the *SmartBoard* from Smart Technologies [88], that is able to do audio recording.

### 1.5.3 Tivoli/MeetingBoard

The objective of *Tivoli* [PMMH93] from Xerox PARC is to support informal business meetings of small groups on an electronic whiteboard. It handles pen-based writing, drawings, wiping, and gestural editing. *Tivoli* has grouping techniques based on automatic recognition of implicit structures and a flexible re-grouping mechanism [MCvM97]. The interface is optimized for sorting, categorizing, and annotating whiteboards content. Board strokes are represented as splines. Gestures are used to select, move, and change the properties of objects shown on the board and to zoom in or out. Scripts can be plugged in for additional stroke processing [MvMC98a, MvMC98b, MvM00]. Various people

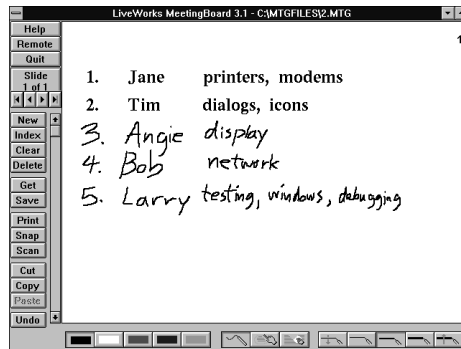


Figure 1.7: Snapshot of the Xerox LiveWorks *MeetingBoard*, the commercial version of *Tivoli*. Picture from [MCvMK95].

can use the system by simultaneously running connected copies, allowing for distributed work.

The original *Tivoli* system was developed for Sun-based *LiveBoards* and Sun workstations. A version named *MeetingBoard* was marketed for commercial *LiveBoards* controlled by Windows PCs, see Figure 1.7.

#### 1.5.4 Flatland

*Flatland* [MIEL99] is another development of Xerox PARC for digitizing office whiteboards. Like *Tivoli*, it is a collaborative application for informal office work. Its design is based on a two-button event model: a standard pen-down on the board surface is translated into a stroke, pressing an extra button triggers a pie menu<sup>19</sup> for controlling the application. The input space is organized into pages, using a flip-chart metaphor. Sessions are stored automatically [IMEL99].

The basic primitives of the board content are pen strokes, which are automatically grouped together based on spatial proximity into “segments” by using their bounding boxes [IELM00]. The board elements can be dragged around or automatically reordered. It features automatic shrinking of segments when squeezed to screen border (“screen real estate management”) and a layout control to prevent overlapping. Segments can be applied “behaviors” to be triggered by the context menu, for example modifying to a map-like output replacing single strokes by double line “streets”, adding check boxes to lines of written text, straightening line segments of sketches, or a calculator behavior processing inputs of numbers and basic calculations. New behaviors can be plugged into the system [MIEL00]. See Figure 1.8 for examples on behaviors.

*Flatland* also has a powerful multi-undo mechanism that allows rollback in time. Each of the segments has an individual time-line stored with a special transaction model, allowing undos and redos based on the segment’s history and independent of the other board objects [EILM00]. The system was implemented in Java.

<sup>19</sup> *Flatland* uses the variant of pie menus with the homing strategy for uninterrupted input sequences. See Section 1.4 for a short description of pie menus.

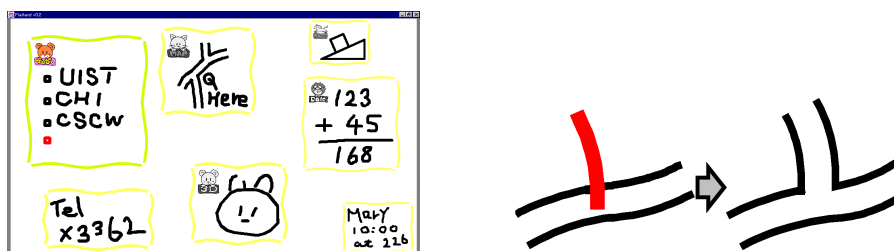


Figure 1.8: Left: Several *Flatland* segments with different behaviors applied to them. Right: Adding a single stroke to a segment with the map behavior. Figures from [IELM00].

### 1.5.5 i-LAND and FLUIDUM

The Fraunhofer *i-LAND* [SGH98] project is inspired by augmented reality and Ubiquitous Computing. Computer-augmented “roomware” is created by integrating computer-based information devices into parts of a room (e. g. furniture, doors, walls) and to form co-operative environments, rooms, and buildings. The project tries to create the “*workspaces of the future*” [SGH<sup>+</sup>99] as an interactive landscape for creativity and innovation, for example group work and informal meetings where participants are at different locations. The controlling *beach* software is written in Smalltalk. [STMTK01].

Apart from a mechanism for establishing relations between physical objects and information objects called the *Passage mechanism*, the project created an interactive table (called *InteracTable*), computer-enhanced mobile and networked chairs called *CommChair*, and an interactive electronic wall called *DynaWall*. This is a 4.5×1.1 m back-projection wall with a resolution of 3072×768 pixels, realized as a combination of three Smart rear projectors, each of the three segments using its own computer. Its software supports video-conferencing and sharing content material between the local and the remote meeting sites. As described in [Gei98], it also supports gestures for

[...] *shuffling display objects around, throw them to office users standing at the opposite side of the wall, [...] take objects and put them back at another location*

exploring new forms of interaction with electronic boards.

The *FLUIDUM* [27] (flexible user interfaces for distributed ubiquitous machinery) lab research at Universität des Saarlandes is of a similar vein, with the goal of developing techniques and metaphors for differently-scaled Ubiquitous Computing scenarios, like interactive desks, rooms, and buildings. Among others, the *wipe gesture* is researched as an interaction technique.

### 1.5.6 IRoom/I-Workspace

The Stanford *I-Workspace* [WJF02] is another project on interactive workspaces. Their *IRoom* (interactive room) is equipped with a display-enhanced table, and a high-resolution back-projection wall, the *Stanford Interactive Mural*, measuring 6×3.5 feet with a 64 dpi resolution and realized with a 4×3 array of projectors.

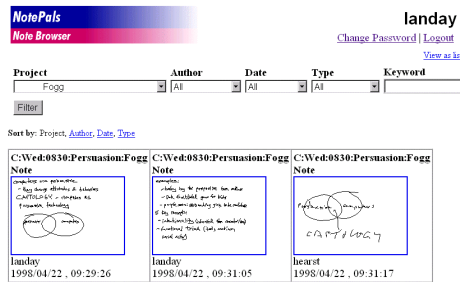


Figure 1.9: The Web-based *NotePals* Note Browser, showing thumbnails of the notes matching the query. Picture from [DLC<sup>+</sup>99].

The *IRoom* allows visitors to interact with the room components via PDAs connected with PPP over serial cable or with WLAN [FJHW00]. They can control the room light, the projectors, PowerPoint presentations shown on the Mural, annotate them (called *Smart PowerPoint* by the developers), and share data objects and applications. In contrast to the *i-Land* project, the *IRoom* strategy is to support standard Unix or Windows applications as regular components of the environment.

For managing the communication between different components, the Java-based room display manager uses a special *event heap* transmitting the events to all receiving clients. The transmission strategy is specially designed for robustness [Win01a].

The *IRoom* applications and interaction schemes researches focus on CSCW tools, brainstorming, and direct interaction with board elements, mainly modification of size and position. Their preferred user interface metaphor are a pen and virtual sheets of paper [Win01b]. To achieve what they call a *fluid interaction* using pen input, the command mechanism relies on *flow menus*, a variant of pie menus [GW00, GMW01].<sup>20</sup> The advantage of flow menus is that they avoid the necessity of visual feedback once learned as stroke gestures by the user and thus reduce the cognitive involvement in submitting the commands. Another concept used is the *typed drag and drop*, where board objects can be assigned types to determine their behavior when dropped onto another typed object. The interface allows continuous freehand scaling, and all actions are logged to supply an infinite undo mechanism. Visual snapshots can be taken [GSW01]. Applications developed for testing the concepts are a brainstorming tool called *PostBrainstorm* [Gui02] and the *Geometers Workbench* [GWW00] for differential geometry by informal sketching, using Wolfram Mathematica for the calculations.

### 1.5.7 Other CSCW Systems

*DOLPHIN* [SGH<sup>+</sup>94] is a system to capture informal work-group meetings, supporting synchronous and asynchronous settings. It manages both shared and private documents. Text, images, handwriting, and audio can be transmitted

<sup>20</sup>See Section 1.4 for a short description on pie and flow menus.

synchronously. Asynchronous access to a meeting's documents is allowed, but the interaction is not recorded for replay.

*Coral* [MHJ<sup>+</sup>95] is a “confederation” of tools to support real-time capture and replay of free form meetings. It captures materials shown on the whiteboard including freehand markings and the audio stream.

The *transBOARD* [IU97] is a physical whiteboard with pen-tracking based on infrared laser scanning. Remote users can view whiteboard activity both synchronously and asynchronously via a Java Applet.

The CSCW system described in [Rek98] combines one PalmPilot PDA per user and a single shared whiteboard. The Palms, called *M-Pad* by the authors, serve as tool palette and data-entry palette for the whiteboard, implementing an oil-painters-palette-and-easel metaphor.

*NotePals* [DLC<sup>+</sup>99] is a note-sharing system for work groups running on PalmPilots. It records timestamped handwritten notes and adds some context information, the name of the author and the project name. In an optional post-processing step, pattern recognition is applied to the notes to support additional retrieval methods. See Figure 1.9.

*DUMMBO* (dynamic ubiquitous mobile meeting board) is intended to capture informal meetings. It records board history and audio as WAV. The recording starts automatically when more than two people gather at the board or when somebody starts to write or erase on the whiteboard [BAT99, DSA01].

## 1.6 Course-Authoring Systems

Course-authoring systems are tools for the non-live development of e-learning material. A number of authoring systems for general HTML page building or for interactive multimedia applications are often used for generating e-learning modules. Commercial examples of the first type include Microsoft *Frontpage* and MacroMedia *DreamWeaver*, the second type includes *ToolBook* [89], *SWISHmax* from SWISHzone [90], *MacroMedia Director* [59], and *AuthorWare*<sup>21</sup> [58]. Most multimedia authoring tools require specific runtime environments or browser plug-ins for replay (for example *Authorware Player*, *Flash Player*, *ShockWave Player*), others can create native applications, e.g. *Multimedia Fusion* [15] can compile its animation to Windows applications.

In addition to these general multimedia authoring systems, there is a range of systems that focus on generating interactive educational material on specialized subjects, for example *Cinderella* [RGK99] [11] for Interactive Geometry and *JavaView* [PKPR02] [45] for 3D geometry viewing and mathematical visualization, both producing using Java Applets. Another example is *Flash-dance* [Esp04], creating Flash animations for algorithm visualization.

The following sections describe a number of authoring systems focusing on the production of courseware for university teaching.

### 1.6.1 MANIC

The *MANIC* [SSL<sup>+</sup>97, PK99, BKT02] [60] (multimedia asynchronous networked individualized courseware) system delivers slides as HTML documents and GIF images, optionally with timed highlighting effects. This is synchronized with an

<sup>21</sup>MacroMedia AuthorWare is even specifically targeted at authoring e-learning content.



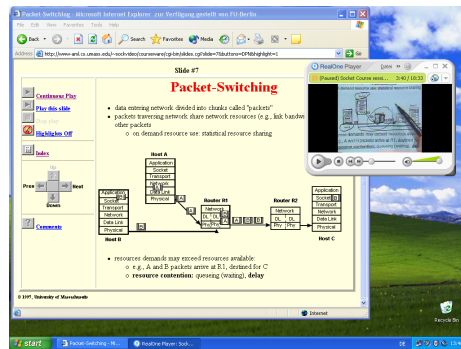


Figure 1.10: A *MANIC* lecture on Unix network programming with RealVideo and RealAudio.

audio stream encoded in RealAudio, experimentally also with RealVideo, see Figure 1.10. The audio is encoded for low-bandwidth quality, requiring only a modem connection. The system also comes with a connection to a MySQL database for storing and searching courses, with the access interface being built as PHP scripts for Apache Webservers. The *MANIAC* 2.0 server is written in Perl, the previous version (numbered 1.5) was implemented in Java.

To create a *MANIC* course, the author creates the HTML pages and/or images externally. To highlight text in the HTML pages, the author manually adds a special *MANIC* tag into the document's HTML code. Audio is encoded from DAT or videotape recordings. The *Timing Data Recorder (TDR)* application creates program timing events for the presentations. At the server, the replay is controlled with CGI scripts. At the client side, a JavaScript-enabled browser with cookies enabled and the RealAudio plug-in installed is needed to view the course. The viewer's navigation options include starting and stopping the replay, jumping to the next or the previous slide, and random access of slides via a table of contents.

### 1.6.2 Cornell Lecture Browser

The *Cornell Lecture Browser* [MS99] [17], which is also used by Berkeley as the *BMRC Lecture Browser* [5], shows slides with a synchronized RealAudio and RealVideo stream. It uses JavaScript to pre-load the slides into the browser cache to reduce delays in slide transitions. All slides have to be converted manually by the author to a browser-supported image format.

### 1.6.3 AudioGraph

*AudioGraph* [JSS98, Jes00, Jes01, Jes03] [2] is an authoring system combining images, lecturer annotations, highlighting of selected areas, and playback of sound clips. The author sequentially adds elements like hand-drawn images or imported images to a slide and determines the transition times between the elements, meaning that the elements appear one after another. He can also add audio to be played, the length of the audio chunks usually determining the transition times between graphical elements. The system generates one

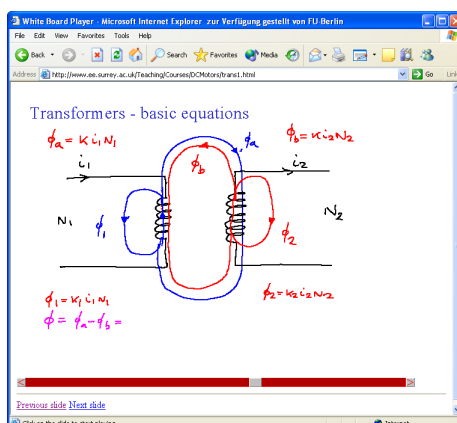


Figure 1.11: An AudioGraph lecture on transformers [61] replaying a handwritten slide.

HTML page per slide and the user can navigate between the slides. Within one slide, the user can start the slide animation from the beginning as well as pause and resume the animation. A progress bar shows the position within the slide animation, but random access is not supported. AudioGraph is designed for low-bandwidth requirements. The audio stream is encoded using GSM 6.1 mobile telecommunication compression.

Figure 1.11 shows a snapshot from a purely hand-drawn AudioGraph lecture, using a Java Applet for playback. The replay of more recent versions of AudioGraph lectures requires a custom browser plug-in to be installed, available for Macintosh and Windows PC browsers.

## 1.7 Presentation and Classroom Enhancement

This section describes systems to assist lecturing in the classroom and tools for live transmission of classes, which do not aim at recording the experience.

A common approach for lecturers who want to transmit a class is to use a video-conferencing solution. However, video-conferencing systems have not explicitly been created for teaching. Their conception assumes symmetric communication and relies on all participants to have equivalent hardware. Great effort is spent transmitting audio and video, but support for the transmission of teaching-specific content, such as board drawings, is lacking. Using standard video-compression approaches, they exhibit the quality problems discussed in Section 1.1.

The most prominent presentation tool is PowerPoint, which has been already discussed in Section 1.1. Demands to augment slide shows by freehand annotations were often expressed to make the talks more flexible than it is the case when using prepared slides only. To this end, several research systems had been devised and by now, Microsoft has added some annotations functionality to PowerPoint.<sup>22</sup>

<sup>22</sup>The *Office XP for Tablet PC (Tablet Pack)* adds “ink” features to Office applications. In

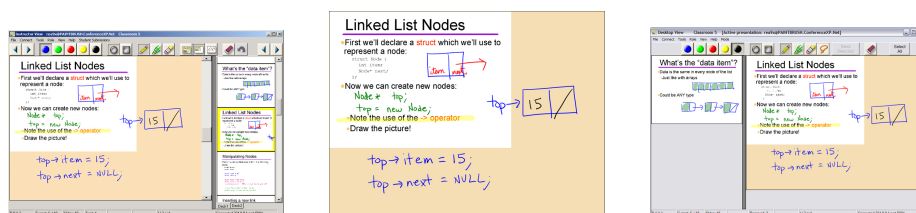


Figure 1.12: *Classroom Presenter*. Left: View of the instructor's screen. Middle: View projected for the audience. Right: View for remote students. All three images from [13].

An early system to integrate prepared slides, handwriting of the instructor, and student contributions submitted from their desks is the *Lecturer's Assistant* [BP94].

The *Pebbles* [MSG98, Mye00, Mye01] project uses PDAs for controlling presentations including stylus-based drawing.

*ActiveClass* [RSTG03] uses PDAs to promote interaction by allowing students to submit text-based questions to the instructor and to respond to polls submitted by the instructor.

### 1.7.1 BIRD Note-taking System

The *BIRD* (*beacon-identified realtime display*) *Note-taking System* [30] is an application based on .NET that allows the lecturer to use annotated slides. The students use laptops with wireless access to the same subnet as the instructor uses and use a client software installed on their machines to receive the lecture data. That software also allows them to save snapshots of the received screen content.

### 1.7.2 ConferenceXP Presenter/Classroom Presenter

The *ConferenceXP Presenter* or *Classroom Presenter* [AAS<sup>+</sup>04] [13] is used for both classroom teaching and synchronous distance teaching and was first deployed in summer 2002. The Presenter is designed to run on Windows Tablet PCs.

In lectures, it organizes content in pages: slides are shown by the lecturer and can be annotated. To provide a kind of whiteboard functionality, the instructor may insert empty slides to be filled with freehand writing and drawings.

For distance teaching, the system is used by a video conference in connection with ConferenceXP [16]. The transmission is two-way and a client needs to have *Classroom Presenter*, ConferenceXP, and PowerPoint (XP or 2000) installed. The data transfer is built on top of multicast communication, which does not guarantee delivery. The developers report difficulties to transmit large objects, such as slides, especially in a wireless environment [13]. As a workaround the application allows a manual re-broadcast to be triggered by the client.

*Office 2003*, these are already integrated.

The system produces three different views: the instructor’s desk, including special controls, the image sent to the projector in classroom teaching, maximizing the current slide with its annotations, and a third view for remote access, see Figure 1.12.

While being designed for mobile use, according to a survey this feature did not rank highly among instructors [AAS<sup>+</sup>04]. Instead, the preferred use was stationary.

Students’ contributions to the common display are usually done by the lecturer passing his or her tablet around, see [AAS<sup>+</sup>04], but experiments with real-time feedback from remote participants as collaborative work in the classroom have also been done. To this end, students connecting via a Tablet PC may send annotated slides to the instructor, who can choose to present those slides and annotate them him or herself [AAV<sup>+</sup>03, SAHS04].

While the setup in connection with the ConferenceXP application would allow to encode the transmitted session to a (huge) stream for replay with Windows Media Player, the reported use of the *Classroom Presenter* so far is exclusively in synchronous teaching.

At sub-projects, the annotations and slide transitions were stored and a custom replay tool was built, albeit exclusively for post-analysis purposes. The annotation patterns of lecturers were analyzed [AHP<sup>+</sup>04b] and the effectiveness of handwriting recognition for this kind of content was studied [AHP<sup>+</sup>04a]. See Section 6.8 for a short outline of the results.

## 1.8 Lecture Recording

*AutoAuditorium* [Bia98] [3] from Telcordia Technologies (formerly named Bellcore) is a system for automatic camera control in transmission and storage of classroom presentations. *AutoAuditorium* setup uses three cameras, one having a front view on the lecturer on a stage, one fixed at a projection screen for a slide projection, and a third looking at the lecturer from a side view. Multiple microphones are used, usually at the stage, in the auditorium and optionally a wireless microphone worn by the speaker. The system mixes the sound signals, tracks the lecturer with the cameras looking at him or her and uses a heuristic approach to switch between different cameras. For example it shows the view on the slide projection when a new slide appears.

*We-Met* [WRZO91, WRB92] (window environment – meeting enhancement tool) is a tool for an electronic whiteboard to show slides with pen annotations. Interactions are stored for replay, but audio capturing is not supported. Handwriting is saved as timed vector data, keeping its storage size small.

### 1.8.1 Desktop Grabbing

Applications that grab desktop activity and store them as video for replay can be used to record lecture presentations, especially if they allow to record the audio of lecturers narration and have tools for adding annotations. For example *Camtasia Studio* [8] by TechSmith Corporation [92] is a commercial Windows 2000/XP application for real-time screen recording and synchronized audio. It also allows place marks with cursor-highlights, and, in a post-production phase, captions, text boxes, and graphics can be added. To avoid the detrimental

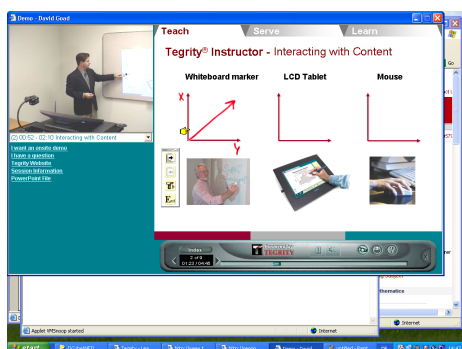


Figure 1.13: Replay of a *Tegrity* recording about using *Tegrity*.

effects introduced by lossy video encodings, a custom lossless codec, the *Tech-Smith TSCC Codec*, was developed. However, it requires the remote viewer to install the codec for replay, which is only available for Windows. The presentations can be exported to standard video formats (AVI, QuickTime, RealVideo) with standard encodings, but this re-introduces the problems of hard-to-read contents.

Another commercial example is *Tegrity* [93]. It captures the screen, provides software for freehand annotation, and records the audio and/or video of the instructor. A Java Applet player handles both live transmission and replay from an archive. The player can be used by the students to add their annotations to the recording, as freehand or as typed text. Figure 1.13 shows a replay of a *Tegrity* recording in a Java-enabled browser.

The *Transparent TeleteachingTool* [ZS02] (TTT) is a screen-recording tool for teaching developed at Universität Trier. It uses the *Virtual Network Computing* [RSFWH98] (VNC) tools for a networked screen capturing of the presentation computer at the TTT server.<sup>23</sup> The captured screen, audio, and an optional video signal are encoded by the server, stored for replay, and multicasted to remote viewers.

In synchronous access, the audio signal is transmitted as uncompressed  $\mu$ -law mono, the video as M-JPEG, encoded using the Java Media Framework (JMF). The screen image is transmitted using VNC's *Remote Framebuffer Protocol* (RFB). For supporting multicast, TTT modified the protocol to use UDP instead of TCP. To compensate for UDP not offering reliable transmissions, TTT added information transmitted for redundancy in critical updates. Bandwidth requirements are about 50-60 kbps for audio, an average of 50-60 kbps for the screen, with peaks estimated to be twice as high, and the optional video ranging from hundreds to over 1,000 kbps, depending on the video stream quality. In the recorded lecture, the video is stored in QuickTime format using the H.263 [Zhu97,BCD<sup>+</sup>98] codec, the audio is stored as  $\mu$ -law mono. The VNC screen stream is directly stored as it is transmitted. Examples can be found at [95].

<sup>23</sup>In [ZS02] the authors report difficulties with handling the audio recording and the transmission on the same computer. As a workaround, they use different servers for the two tasks of archiving and live transmission.

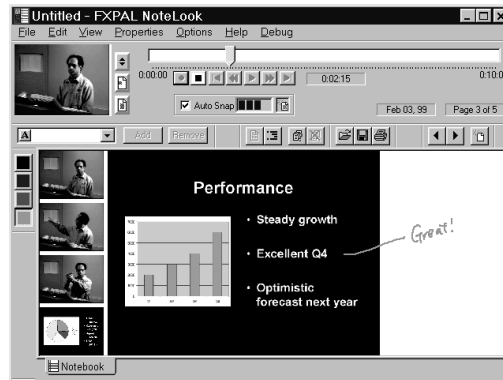


Figure 1.14: The *NoteLook* client for synchronous access. Figure from [CKRW99].

The client for both live transmission and replay from recording is realized as a Java application, providing platform-independent access. The client application requires JMF to be installed.

### 1.8.2 Just-In-Time Lectures

With Carnegie Mellon's *Just-In-Time Lectures* [47] (JITL), the instructor's teaching is recorded on analog video. Later this is digitized and encoded as a QuickTime movie. Slides are stored as high-resolution images and synchronized with the video stream by the *Just-In-Time Lecture Player*. An additional textual sections overview is added to provide contextual navigation. The system uses e-mail for questions to the teachers and access to the online library (a FAQ) as "interactive" features. Disk space requirements for the streams are quite high [DC97].

### 1.8.3 NoteLook

*NoteLook* [CKRW99] is a *Dynamite*<sup>24</sup> successor, but with the focus on note-taking in a shared lecture instead of on office work. It allows synchronous remote access to the lecture and uses the student's control for capturing individual recordings. The students are equipped with wireless connected pen notebooks, and can switch between different video channels (usually takes on the room or on the presentation material). The selected video stream is stored for replay and single video frames can be grabbed and annotated. It also features a mode for automatic grabbing when the video image changes, which can be used for detecting slide transitions. See Figure 1.14. The system creates (static) HTML pages with the grabbed snapshots of the notes. Selection of an image starts the video playback beginning at the snapshots time offset.

<sup>24</sup>*Dynamite* is described in Section 1.5.1.

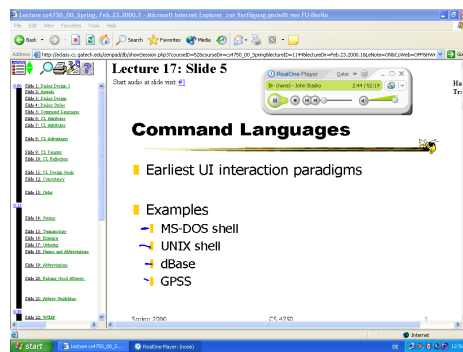


Figure 1.15: A replay of an *eClass* lecture with slides and audio.

#### 1.8.4 DEBBIE/DyKnow

With the *DEBBIE* [BJJ01, BBW04] (DePauw electronic blackboard for interactive education) system and its commercial offspring *DyKnow* [20] (dynamic knowledge transfer), students are equipped with PCs with “video tablets” (Tablet PCs or PCs with digitizer tablets with built-in LCD screens). The instructor uses an electronic whiteboard or also a video tablet. The system allows to present prepared material, keyboard typing, and freehand sketches. The material is transmitted over a network connection to all student workstations. Students can add their own annotations, both freehand and typed. The students may send portions of their work space to the teacher, who can integrate them in his or her presentation for the entire class to see. At the end of the class, students can save their workspace for static viewing, printing, and replay. Audio capture is not supported.

The *DyKnow* Implementation is based on Microsoft .NET Framework and runs under Windows 2000 and XP.

#### 1.8.5 Classroom 2000/eClass

Georgia University developed a technology originally named *Classroom2000* and later renamed to *eClass* [Abo99, Bro01] [14]. It records lecture slides and handwritten annotations as static Web pages together with audio and (optionally) video recordings of the instructor and any Web page he or she visited during the lecture.

The presentation component *ZenPad*<sup>25</sup> handles showing slides in GIF or JPEG format, added beforehand in a pre-production phase, as well as freehand annotation. “Empty” slides can be added during the presentation to be used as a whiteboard. Any Web page URL the lecturer visits is recorded by having the demonstration Web browser use a custom proxy server. Audio and low-quality video of the lecturer are encoded in RealVideo and RealAudio.

<sup>25</sup> *ZenPad* is a pure Java successor of the original *ClassPad* program written in Visual Basic. The *ClassPad* application produced static annotated slides, but involved manual upload and structuring of lectures with about four man hours per lecture. *ZenPad* was developed to have an automatic upload and to support certain replay features.

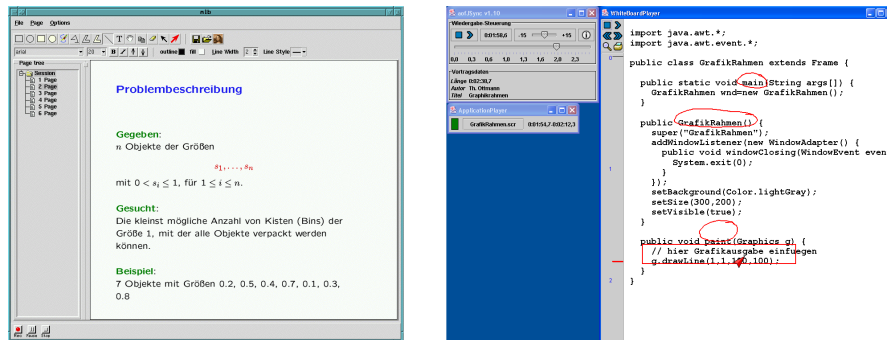


Figure 1.16: Tools from the *AOF* package. Left: The *mlb* tool for recording lectures, image from [1]. Right: A recorded lecture replayed by the Java implementation of the *aofSync* player.

The *ZenStarter* program is used to start different *eClass* system components using different options. Starting *ZenPad* triggers the *ZenStarters* for audio and video encoding. After the recording session, the *StreamWeaver* application is used to build HTML documents integrating the timestamped streams. Also, a list of links to each annotated slide as well as links to the encoded audio and video streams and visited URLs is produced. The annotated slides are transformed to GIF and JPEG images. Post-processing is reported to take about one minute for a complete lecture. The generated material is stored in a database. The HTML pages generated from slides have a size of about 20-200 kB per lecture, the audio stream 3-5 MB and the video 5-7 MB.

For replay, the viewer can browse through the graphic slides and choose the Real stream to jump to the time position of a slide. See Figure 1.15 for an example.

The *StuPad* [TA99, TAB99] (students notepad) extension of *eClass* broadcasts the instructor's writing to the students' desk and allows them to add private notes.

### 1.8.6 AOF

*AOF* [1] (authoring on the fly) is a system for both synchronous and asynchronous delivery of lectures. Transmission includes whiteboard activities, in most cases annotated slides and audio, optionally also video and animation applications. *AOF* lectures are usually given using electronic whiteboards or digitizer tablets.

For asynchronous delivery, the client has to download all the presentation material. Streaming is only supported for the synchronous scenario. Synchronous delivery is based on the MBone [Eri94, Cla98] technology. Originally, it directly used *vic* (video conferencing tool), *vat* (audio conferencing tool), and *wb* (shared whiteboard tool) from the MBone tool set for showing PostScript slides. These tools allow for paging up and down plus simple drawing functionalities. The *AOFwb* [BO96] was developed for several Unix platforms to replace *wb* for more features. Its successor *mlb* (media lecture board) shown in Figure 1.16 is available for Microsoft Windows and handles drawings as vector graphics.



It can import PostScript and PDF documents (using *Ghostscript* [28]), images (GIF, JPEG, TIFF, XBM) and MPEG-1/-2 streams. Audio is transmitted and stored as uncompressed PCM (.aif).

The `aofRec` application is used as a synchronous client. For replay of recorded lectures, a different client program named `aofSync` is used, see Figure 1.16.

For optional post-production, an editing tool named `AOFedit` is available for Unix. The *AOF* system also features a number of other tools for preprocessing, like the PowerPoint importer application `ppt2aof`, and for post-processing, like the indexing tool `aofSE`.

### 1.8.7 Lecturnity

*Lecturnity* [50] is a commercial lecture-recording tool by *imc Advanced Learning Solutions* [37] written in Java and run on Windows.

The *Lecturnity Assistant* uses MS PowerPoint slides and images (BMP, GIF, JPEG) as the basis of a slide recording to be captured with annotations. The presentation can also use a screen-grabbing functionality. Annotation tools include pointer, marks, freehand drawing tool, a highlight tool, and typed text. The presentation is captured together with an audio stream and can optionally be combined with video. The capture application runs on Windows, using the Java Runtime Environment, the Java Media Framework (JMF), and the Java Advanced Imaging (JAI) library. Video is encoded with the Indeo Video 5.0 codec.

The *Lecturnity Player* for replay of *Lecturnity* recordings is a Java application. A presentation can also be exported to RealMedia or Windows Media formats by the *Lecturnity Publisher*. This allows replay in browsers with JavaScript, StyleSheets and *RealPlayer* plug-in or Windows Media plug-in.

The *Lecturnity editor* for optional post-processing allows to cut and paste parts of the recording and to combine them with external audio and video streams.

### 1.8.8 Tele-TASK/t-Cube

*Tele-TASK* [CMMS03, MSCM03] [94] (tele-teaching anywhere solution kit) is used for synchronous and asynchronous transmissions of lectures. It delivers RealAudio and RealVideo of the lecturer plus a sequence of screenshots – normally of an annotated slide show by the lecturer. The slides are captured at one frame per two seconds. For live streaming, the slides are streamed as BMP images, requiring the remote viewer to use a high-bandwidth connection. In a post-processing step, a narrow-bandwidth version (ISDN or modem) is produced by encoding the snapshots as PNG graphics, utilizing PNG transparency to achieve an encoding into difference and key frames. For replay, the *RealPix* technology from RealNetworks [79] is used to stream the slide images and SMIL for synchronizing the different streams.

The system uses an extra PC for capturing and encoding. The solution is available commercially as all-in-one hardware to be connected to the presentation computer under the name *t-Cube*.<sup>26</sup>

---

<sup>26</sup>Note that *t-Cube* is unrelated to the pie-menu based key input system *T-Cube* described in Section 1.4.

## 1.9 Conclusion

The approach presented here tries to solve the problem of huge costs for creating courseware material by generating it as a by-product of regular classroom teaching. The chalkboard lecture becomes a form of e-learning authoring. The approach benefits from many professional lecturers being able to simply walk to the chalkboard and start up a spontaneous talk with a high degree of learner interaction. A good chalkboard lecture should automatically result in a good e-learning lesson. The didactic advantages and the easy handling of the traditional chalkboard should be preserved while extending its reach to distance learning. Storage and transmission of the lecture must be realizable with negligible additional effort.

In the classroom, the system tries to enhance teaching quality by enabling the instructor to integrate multimedia elements. Students should be given the time to grasp the concepts. After the course, they have access to a recording of the lecture with all relevant information. To minimize barriers for first-time users, they should be able to replay the recordings using just a standard Web browser, without requiring them to install any special software.