Chapter 2

User Interface

2.1 Overview

In the Lecture Room

The E-Chalk system starts with a setup dialog for the recording settings. In most cases, the instructor just keeps the settings that were stored from the previous session, changing just the entries for the lecture title and the path to store the lecture to. After hitting the *OK* button, the system's user interface metaphor changes from the computer desktop to the chalkboard, a display to be shared with an audience. The E-Chalk board window occupies the whole screen. The instructor draws and writes on the board with a pen-like input device.[1] A tool-box type dialog allows to change drawing colors and widths. It also provides access to a number of multimedia elements, like adding images from a local hard drive or the Internet to the board, or integrating interactive animations.

The board can be scrolled up and down vertically, providing the lecturer with a virtually unlimited surface to write on. Instead of using a desktop-style scrollbar, two *drag handles* are provided at the top and at the bottom of the screen. The user grabs the board at a drag handle with the pen and drags the board up or down.

All actions on the board are tracked and recorded. The development of the board content can be viewed by a remote learner, both as a live transmission and as an asynchronous replay. The voice of the lecturer is recorded along with the board stream. These data streams already capture most of the teaching substance. Optionally, a video stream of the instructor can be added to give the remote lesson a more personal touch.

The system does not require the user to explicitly trigger a save. Everything is automatically stored for viewing with a Web browser.

Remote Access

When remote students open E-Chalk's generated Web page of a given course with a browser, replay starts in the form of self-synchronizing Java Applets. One

[1] Different hardware solutions for pen-like input devices and big displays are described in Section 8.1.

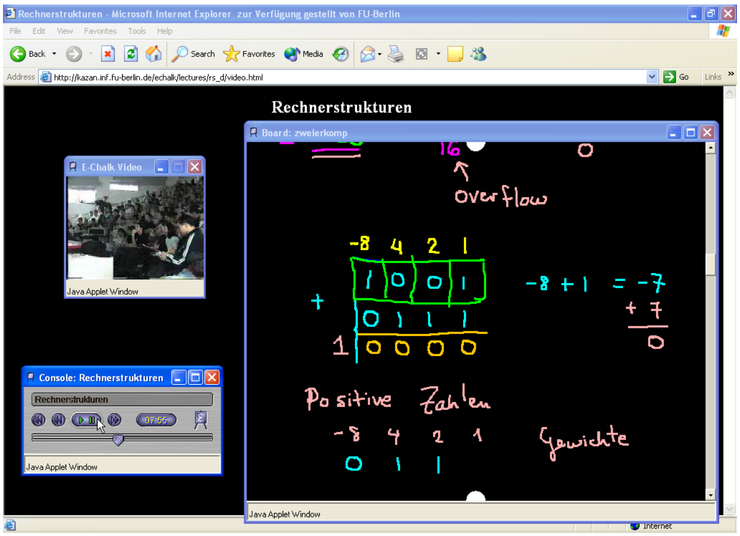

Figure 2.1: Replay of the first E-Chalk lecture recorded [85] in a browser.

Applet is started for each recorded data stream, board, audio, and video. Another Applet, a control panel, is provided for VCR navigation through archived lectures, see Figure 2.1. All these Applets run in a standard Java enabled browser[2], without requiring the user to download a plug-in, and this solution is completely platform independent.

The required bandwidth for the transmission of audio and dynamic board content for the setup with the default audio quality is 64 kbps. The network traffic generated by the board content is negligible compared to the traffic needed by the audio signal, since the board uses a vector format. Using a video stream requires a further 64 kbps.[3]

The replay provides the distance learner with a live script where the teacher's side notes are preserved. In addition, a static copy of the final board image as an Adobe PDF file is included for the students to print.

2.2 Usability Considerations

Usability considerations played an important role in the design of the E-Chalk system. Only with an easy-to-use interface, the software can be expected to gain the user's acceptance.

For the interaction on the board, guidelines common for desktop programs will have to be taken with a grain of salt. The board explicitly uses a different interaction metaphor and the guidelines sometimes have to be bent, keeping their underlying arguments in mind. On the other hand, new requirements appear, for example the use of the keyboard is very disruptive when working with a pen at the board and should be avoided as far as possible. Also, double

[2] The Applets need only Java 1.1, to avoid the requirement of a Java upgrade for the browser, and they need not be signed.

[3] Again, the given bandwidth is for the default setup.

clicks should not be required. Some persons already have trouble doing double clicks with mice, and these are even more difficult to do with pen input devices.

Control by Keyboard

Common usability guidelines recommend that an application should be fully controllable with the keyboard, with no need to use the mouse. Reasons are, that expert users will often be faster with keyboard-only inputs and that one should still be able to control the program in the absence of a pointing device. Consequently, the E-Chalk setup dialog and all its sub-dialogs provide mnemonic keys for all input elements.[4]

However, this does not hold true for the main board operations, painting and handwriting. These cannot be naturally performed with a keyboard. The board is meant as an interface for a pen device instead of a keyboard. Still, most non-painting board actions can be controlled via the keyboard, for example `[Ctrl]-Z` is mapped to *Undo*.

In the case of having no mouse or pen device to operate the board, it is important that the standard shortcut for exiting the application, `[Alt]-[F4]`, works on the board. This can be crucial if one starts E-Chalk without a pointing device, or if a battery-powered input device runs out of power.

The keyboard shortcuts are also useful to control the board with configurable hardware devices. Several digitizer tablets and digital whiteboards allow to map key strokes to special regions of the input area. With the board receiving these keystrokes and mapping them to board actions, these serve as buttons for board control. This interfacing technique is also used for the bluetooth hardware extension of the pen described in Section 8.1.1.

Consistency

Consistency is an important aspect in user interface design.[5] This avoids confusion in users and helps them to learn to use the interface quickly. What is also desirable is consistency with other applications – the interface should adhere to accepted standards. Standard GUI elements and symbols as well as standard label names should be used wherever they are appropriate. Various platforms provide guidelines for adhering the the platform's standards [App92, App01a] [29, 49, 68]. Since the E-Chalk software is a multi-platform development in Java, the corresponding guidelines for Java programs [Sun99, Sun01] were used.[6]

Interface Logic

The interface must adhere to the logic of the user's view, not of the internal workings. It has to focus on users and their tasks, not on the underlying tech-

[4] The only input elements without associated mnemonic shortcuts are those operated with the `[Enter]` or `[Esc]` keys, typically *OK* and *Cancel* buttons. Of course, access by controlling focus with keyboard tabbing is also supported.

[5] This is listed first among the "*Eight Golden Rules of Interface Design*" in [Shn98].

[6] For multi-platform programs, some special considerations have to be taken into account. For example, the color reduction algorithms for different OS produce different output. Graphics that look good on one platform may be ugly on another. [Sun99] gives rules for designing graphics like button icons, that will appear acceptable on all platforms.

nology. For example, there is no good reason to require E-Chalk users to trigger an explicit save. Instead, sessions are always stored.[7]

The language in the interface must use clear wording and avoid "tech-speak". Interface texts should be short and to the point since lengthy explanations are rarely read. A common mistake in graphical interfaces is the attempt to compensate for explanations not being read by adding even more text.

Short explanations via tooltips are unobtrusive and should be used wherever possible. For more detailed explanations, a context-sensitive help should be used. Ideally, all texts are localized to the user's language.

Interfaces must heed the fact that "*people make errors routinely*" [Nor88]. With a possibility to undo any action, users are provided with a low-risk environment. It invites them to explore the program and promotes learning.

The users should be guided by the software. Handling errors should be prevented in advance, instead of returning error messages when they occur. For example, when the computer has no sound card, the E-Chalk setup option for recording audio is made inaccessible. The corresponding input element is deactivated and grayed out.

User should not be burdened with technical work which it can be done by the system. Guidance and avoiding an excess of choices greatly disburden users. This is especially important for the board interface, as the operating persons will concentrate on the task of lecturing. Distracting them is very unfavorable. This is an experience also reported with other lecturing tools, for example the developers of the *Classroom Presenter* learned their "*lesson of parsimony*" when they observed that "*instructors were strikingly restrained in their use of Presenter's features*" [AHP+04b].

While users should be guided, they should always be in control. They should be assisted in their tasks, not constrained. Common operations should be quick and easy, while uncommon things should be still possible.

Avoiding Modes

Mode errors occur when the user misclassifies a tool's status resulting in inappropriate actions, for example when one tries to fast forward on a VCR during record mode, or when one turns the car key when the engine is already running. The notion of mode errors was first defined in [Nor81].

Several types of modes are distinguished. *Spatial modes* dependent on location of input tool, like click on which element. The locus of attention is usually at mode determining object and therefore these types of modes are not prone to errors. *Quasi modes* depend on the state of the system and use a kinesthetic feedback to the user. Examples are pressing down the shift key, mouse dragging (with mouse button down), and opening popup menus while holding the mouse button down. Experiments have shown that these modes are also not prone to user mistakes [SKB92]. *Temporal modes* depend on the internal state of the system. Temporal modes increase the cognitive load of users and are

[7] An objection to this might be that it uses disk space, even if the session is only meant for drawing a quick illustration, as E-Chalk is sometimes used in brainstorming sessions. Also, first-time users were sometimes irritated because they were already "trained" by numerous applications to always explicitly save their documents. However, in this case the value of adhering to the standard hardly outweighs the risk of losing recordings of complete lectures, just because instructors forgetting to save them.

error prone. For example the `[Caps-Lock]` key introduces a mode for typing actions. The situation is especially grave when a user is interrupted in his or her action between triggering a mode change and employing it. An infamous example of a mode-heavy interface is the Unix *vi* editor.

To get rid of modes was strongly voiced by [Tes81]. An example of a computer system that tried to avoid modes in the user interface was the *Canon Cat* [Ras00]. When modes are encountered in a group situation on the electronic board, they can be expected to be especially disruptive. The instructor is not only occupied with the board task, he or she also has to keep the audience in mind, and any mode error distracts the lecturer and the audience from the content of the lecture.

While temporal modes should be avoided where possible, it will not be possible to eliminate them fully. When allowing the user to draw with different colors in a painting application or on the E-Chalk board, having the current color as a mode is necessary.

Often the interface can help to avoid mode errors by giving visual feedback like the mouse cursor. This feedback can go unnoticed as the task and not the feedback is in the locus of the user's attention [Ras00]. Feedback works best when it is observed as part of the task itself. For example, in a painting program, the size and shape of the drawing tools can often be chosen. Setting the mouse pointer to the outline of the selected tool can protect from a wrong shape or size setting in drawings, as the user focuses on the position of the very pointer shape when he or she starts a drawing action.

According to experiments, visual feedbacks can help, but kinesthetic ones are much more effective, both in preventing errors and in terms of reducing cognitive load [SKB92]. Important for the effectiveness is the "kinesthetic continuity" by having the user maintain the mode actively. Re-active feedback (like having to hold down a button) performs much better than pro-active feedback (like having to push a button before the action).[8]

Temporal modes are also introduced by *modal dialogs*. They were first introduced by Apple with the Mac OS [App87]. Their purpose is to get the attention of the user by preventing them from interacting with the main application. Unfortunately, this means modal dialogs are highly disruptive. Dialogs change the locus of attention and often hide information on the screen that is needed by the user for the interaction.[9] Modal dialogs are sometimes acceptable to guide users in tasks they triggered themselves. The most disruptive usage of modal dialogs is to display messages, e. g. error messages. This should be employed very sparingly, especially when used to get user input. Frequent use of confirmation dialogs tends to condition the users so as to give a quick answer, even before the request is consciously noticed [Ras00]. For example, requiring a confirmation for each file deletion quickly trains to confirm any deletion without thinking [Nor98]. This does not only fail to protect against involuntary deletion anymore, it also means continually having to do one click more than necessary.

---

[8] Typically, those tasks already occupy the user's visual attention. Visual feedback on the mode has to compete for the user's attention. Using different sensory channels is thus more effective, for example auditory feedback can also reduces mode errors [Mon86].

[9] This observation caused the introduction of semi-transparent dialogs in the Mac OS.

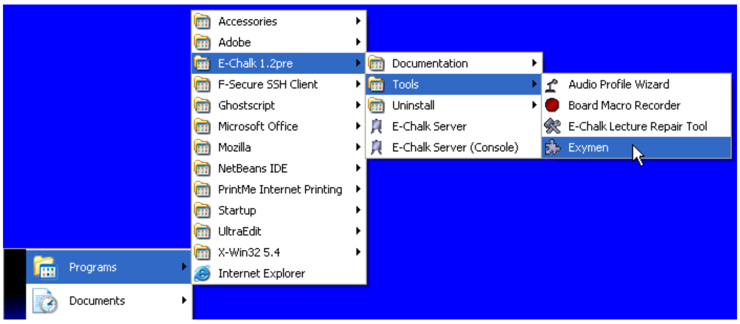

Figure 2.2: The E-Chalk package added to the Windows start menu.

Responsiveness

Responsiveness of applications is essential to avoid user frustration [Joh00]. Giving no feedback for longer durations is very irritating to users. They become concerned that something may be wrong. When an operation takes some time, some sort of feedback should be given. Tolerances are reported to lie in the range of one to three seconds, with most guidelines working with one second [Joh00, Sun01, Shi00]. For prolonged operation in E-Chalk, a feedback dialog is displayed after half a second. If the progress of the operation can be measured, a progress bar is shown. Ideally, the process can be canceled, allowing to discontinue operations taking too much time.

The *perceived performance* of responsive operations is much better than that of non-responsive ones, even when the latter are technically a bit faster.

2.3 Installing E-Chalk

To provide for easy installation of the E-Chalk software, an installer for each supported platform (Windows, Linux, and experimentally also OS X) has been provided. InstallAnywhere [110] is used to build installers for these platforms, setting up the complete E-Chalk software package with all resource and configuration file. This includes the installation of a Java Runtime Environment, enabling the software to rely on a specific Java version to be installed. On Linux systems, the package is registered to the RPM database, on Windows and OS X, the components are registered in the platform application menu, see Figure 2.2.

The installer supports several different languages, localizing both the installation process and the installed E-Chalk application. Currently, only English and German are supported. An older version was also localized to Spanish, and parts of the E-Chalk application have been localized to Turkish. See Section 3.9 for a description of the E-Chalk internationalization architecture.

2.4 The E-Chalk Server Application

When the main E-Chalk application is started, a splashscreen window is shown until the E-Chalk setup dialog appears. Showing this splashscreen gives the user feedback on the applications loading process still running.

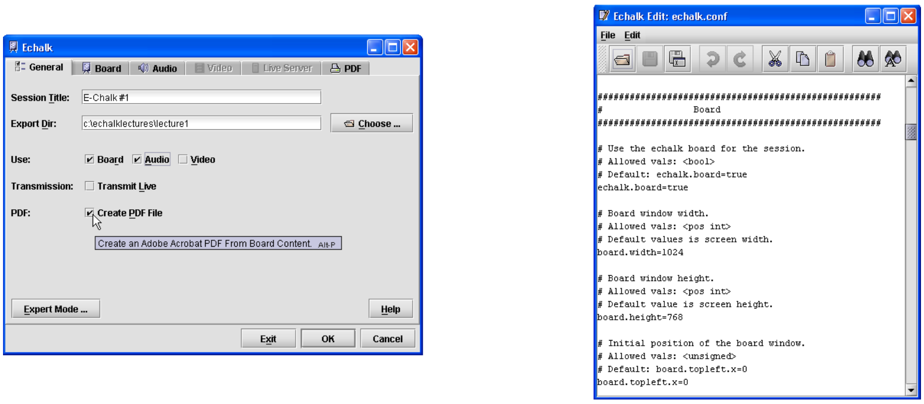

Figure 2.3: Left: Setup dialog for *General* settings. Tabs for *Video* and *Live Server* are deactivated as their corresponding checkboxes are not selected. Right: E-Chalk text editor started for the expert-mode configuration.

2.4.1 Setup Dialog

The E-Chalk setup dialog allows to change the user recordings setting. All these settings are stored persistently in a human-readable format.[10] The different settings are grouped into tabbed pages, with the *General* settings selected at startup, as shown in Figure 2.3. Users can set the lecture's title and the path of the directory to store the recording to. They may select which types of material to store (board stream, audio stream, video stream, PDF from board content) and whether the lecture should be transmitted live. In most sessions, the settings on the following pages can remain unchanged.

The dialog serves as a graphical interface to conveniently change all parameters commonly-used. Some configuration entries, however, are only interesting to expert users. To avoid these settings cluttering the dialog, they are to be changed directly in the file where they are stored. To this end, a button labeled *Expert Mode ...* is part of the general setup page, see Figure 2.3. It is used to load the settings file in a text editor. The editor appears and the dialog is hidden. When the text editor is closed, the dialog re-opens with the newly-loaded settings. See Figure 2.3 for a snapshot of the settings file loaded into the text editor.

On hitting *OK*, the settings are stored and the lecture recording starts. *Cancel* can be used to discard any changed, the *Exit* button stores the changes and closes the session without starting to record.

Each tabbed page of the setup dialog features a *Help* button, which starts a built-in browser on the page's help section of the system's HTML documentation.

Board Settings

The settings controlled with the board page control the size and background color of the board, see Figure 2.4. The width and height values accepted by

[10] See Section 3.2 for details on the format.

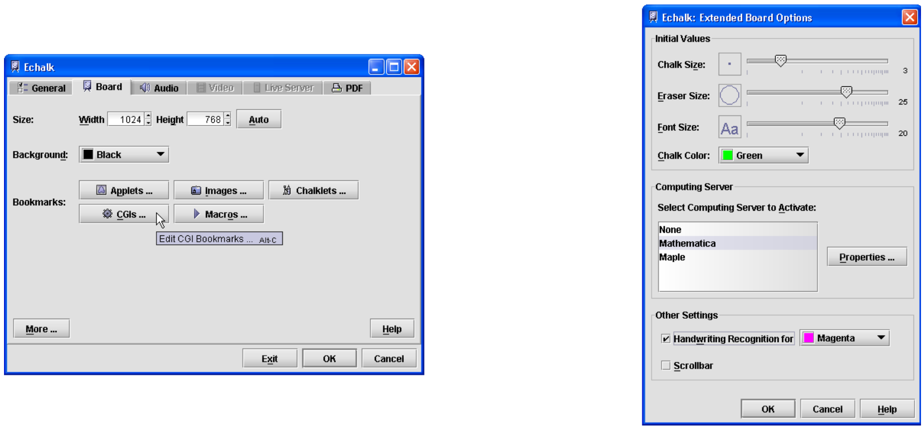

Figure 2.4: Left: The board settings page. Right: Dialog for extended board options, shown from the board's page with the *More ...* button.

the dialog are bound to the display resolution. The *Auto* button sets the size to full screen. Initial values for the board drawing tools (colors and sizes) can be set, with the logarithmic-style size choosers shown identically in appearance in the board interface. The boards mathematical handwriting recognition can be bound to a drawing color, and a computing algebra server program can be connected for use on the board. See Section 4.8 for a detailed description of the handwriting recognition. For details on the setup of computer algebra systems, see Section 3.3.

The dialog allows to choose the board background color among 18 different colors. In practice, however, black is almost always used, with a few users preferring white for similarity with a paper background. Black not only yields the best resemblance with the blackboard, white on black writing is better readable than vice versa, at least for vision-impaired and older people [Gat03, Byr00].[11]

The board supplies access to a number of bookmarks, see Section 2.4.2. These include bookmarks for images and Applets to be pasted to the board, for board macros to be played[12], for CGIs to be queried, and chalklets for interactive board animations. Each of the associated buttons starts a bookmark editor. An editors allow to add URLs and associate labels and icons with them, see Figure 2.5. The editors support editing by drag and drop and allow to import the files of a directory as a list of bookmarks. For macros, replay speed can be set and extra information on the macros is shown, like minimum board width required for replay and background color. The CGI bookmarks have a list of CGI parameters associated with them. Each parameter has a number of entries,

[11]Many electronic magnification tools for vision-impaired users include the option of reverse polarity display, because of negative contrast (white letters on black) being often more readable than positive contrast. Similarly, the Windows magnifier tool optionally allows reverse polarity for accessibility. A known cause for inverse polarity being superior is that physiological changes in older eyes can lead to an increased sensitivity to glare [EW90]. Also, experiments showed that vision-impaired readers with cloudy ocular media perform better with white letters on a dark background [LRPS85].

[12]Board macros are generated with the macro recorder described in Section 6.9

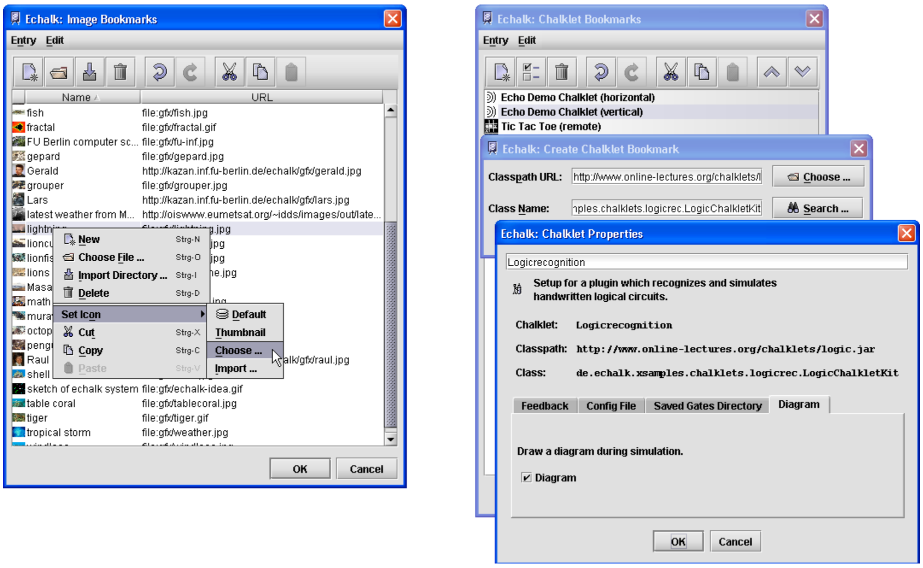

Figure 2.5: Left: The bookmark editor for image bookmarks. Right: The bookmark editor for chalklets in the process of defining a new entry. The properties dialog provides a tabbed page for each of the four expected build parameters.

defining formal name, prompt for query on the board, and HTTP send method (POST or GET).

The bookmark setup for chalklets is even more complex. New entries are created in a dialog. It expects the URL where the chalklet's code should be loaded from and the name of the `ChalkletKit` class[13] to produce the chalklet instance. To make the setup easier, the *Create Bookmark* dialog allows to automatically search the classpath URL for all `ChalkletKit` classes.[14] After that, a dialog showing the properties can be displayed: the meta information defined by the kit is shown and input components for the parameters expected by the kit are provided, see Figure 2.5.

Audio Settings

The audio settings page allows to select a codec to be used for the audio stream, see Figure 2.6. The page also provides a list of audio profiles. An audio profile is used to tune the recording for quality, depending on the information on audio hardware and speakers frequency range stored in the profile. See Section 5.2 for details.

The Button labeled *Create ...* starts the audio profile wizard[15] to generate

[13] See Section 4.6.4 for a technical description of chalklet programming.

[14] An alternative approach would have been to require the chalklets to be packed in a Jar archive together with meta information specifying the `ChalkletKit`'s name. This approach is often pursued in component-based programming, for example for Java beans or for Oscar/OSGi bundles. However, the need for meta information can be confusing for beginners. The required packing also introduces an extra step for developers. This can be annoying when one wants to quickly test modified versions.

[15] The audio profile wizard is described in Section 5.2.3.

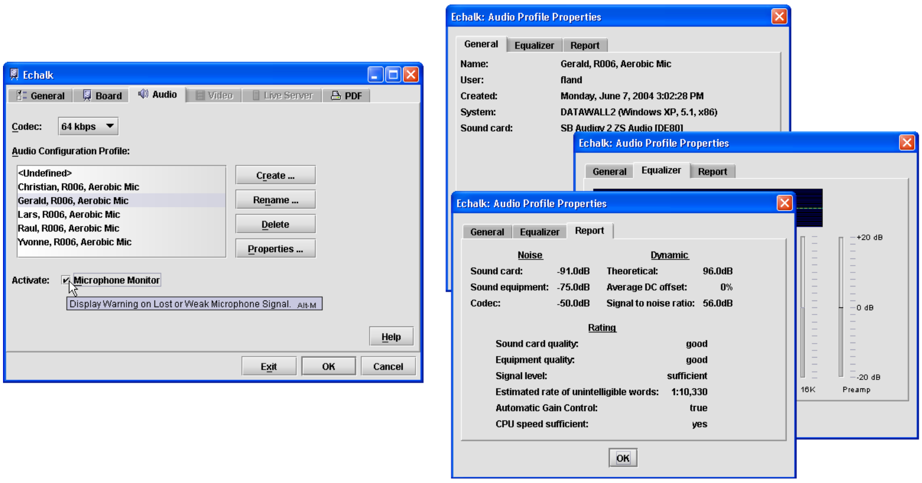

Figure 2.6: Left: The audio settings page. Right: The tabbed pages on the audio profile properties.

a new profile. With a selected profile, the audio settings allow to activate the *Microphone Monitor* during recording, showing a warning dialog when the microphone signal is lost, usually due to batteries running low in a wireless microphone.

To help for remembering which profile belongs to which configuration, one can select to display the properties of the profile. Information about the date the profile was created and the hardware setup are shown in addition to the equalizer settings and the audio wizard's summary report, see Figure 2.6.

Video Settings

In the settings for the video stream, the user can choose the video source to capture, and the resolution and the frame rate to be used. Resolution and frame rate have to be chosen with care with regard to the bandwidth they will consume.[16]

During the installation process, E-Chalk scans for available video sources. As this process takes several seconds, it is not done every time E-Chalk is started. However, the video settings page allows to trigger the scan process manually to make newly installed video sources known to the system. See Figure 2.7 for a snapshot of the video settings page.

Live Settings

In the live settings page, a maximum number of live connections can be defined, depending on the server's performance. The TCP server ports the streams are sent over can be redefined. This is only important if the default ports are already occupied on the server or if their use is disallowed by a firewall.

[16] A planned extension to the setup dialog is to display the average and worst case bandwidth in the setup dialog, both for individual streams and in total.

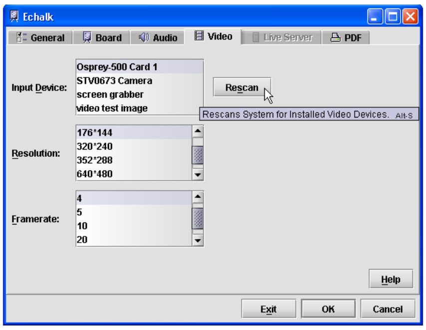

Figure 2.7: The video settings page.

PDF Settings

In the settings for the PDF version of the board lecture, paper orientation and format[17] of the produced PDF can be selected. Regardless of the background color used for the board, the PDF may be produced with white background to save printer color. See Section 6.1.3 on provisions taken to preserve a good contrast of the writing against the white background. Finally, the setup allows to generate a grayscale version of the PDF for black-and-white printing, a color PDF, or both. See Figure 2.8 for a snapshot on the PDF page.

Recording

When the *OK* button is hit to start a recording, the system checks if the output directory already contains a recorded session. In this case, the user is queried if E-Chalk should overwrite the session, continue the old session by appending, or if the recording should be abandoned, returning to the setup for changing the output directory.

Following that, the setup dialog is hidden, and the different recording components (board, audio, video) are initialized for recording. As the setup may take a few moments[18], a dialog with an animated indeterminate progress bar is shown if it takes more than half a second.

During the recording, stream data are written continuously to the disk. This avoids potentially long saving times at the end of a lecture and minimizes data loss in case of an abnormal program termination.

When the recorded streams include the board stream, the recording is ended by exiting from the board interface. Otherwise, the system displays an information dialog with an indeterminate progress bar animation and a button to stop the recording, see Figure 2.9 for an example.

After the recording, a recorded board stream is converted to a PDF transcript, if the option was chosen in the setup. Again, if the process takes longer than half a second, dialogs with progress bars are shown.

[17]The paper formats supported in the current implementation are DIN A0 to A6, letter, half-letter, legal, note, and ledger.

[18]Especially appending to a long previous session may cause the board initialization take some time, because the board must load the complete data of the old session.

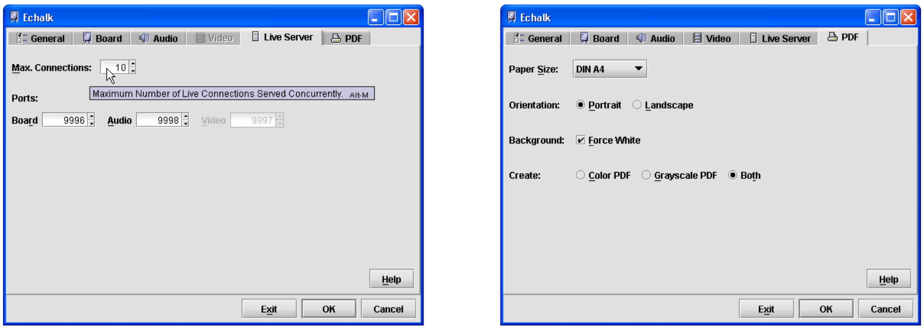

Figure 2.8: Left: The live server setup page. The video server port element is disabled because video recording is disabled in the general settings. Right: The tabbed page on PDF generation.

Figure 2.9: Information dialog during recording when only the audio stream is stored.

2.4.2 The Board Component

Tool Dialog and Basic Drawing

When the board component is started, the user can use the pen device for painting. Color and drawing thickness are controlled with a tool dialog, see Figure 2.10. One can also change from the standard drawing tool (the "chalk") to the eraser tool. It draws with the background color and has the drawing width defined independently from the chalk tool, as a bigger stroke is usually wanted for the eraser. The sliders to control the sizes of the drawing tools (and any typed text) use a logarithmic scale, since fine size control is much more important for the smaller sizes. The mouse pointer on the board is shown as a circle of the current drawing/erasing size or as symbolic chalk/eraser icon, when they become too big[19], see Figure 2.11.

The tool dialog includes buttons to cause undo or redo actions on the board contents. The undo stack is not bound to a fixed number of actions.

A rarely used option of the tool dialog is a button to clear the whole board.

[19]The maximum size of mouse pointer images depends on the platform's window system. On Windows, the maximum image size is 32×32 pixel, on X windows systems, up to 64×64 pixel are allowed.
A possible improvement would be to show the drawing tool pointer in the current drawing color.

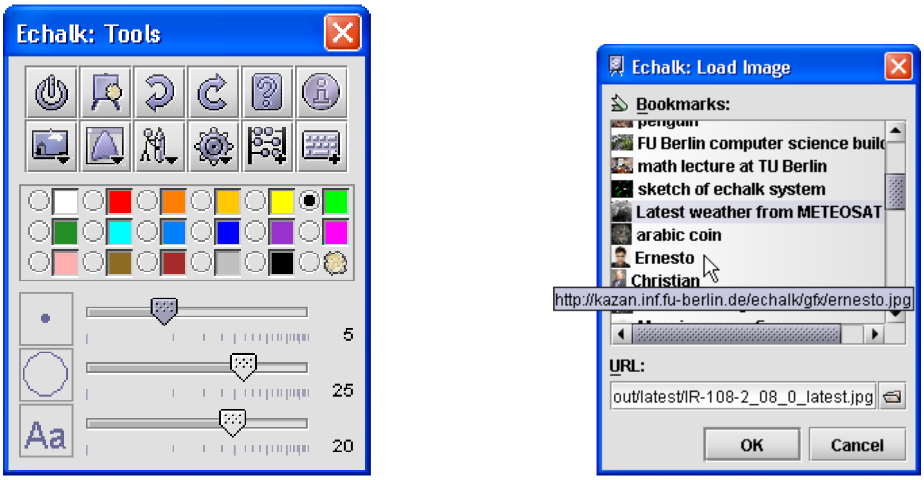

Figure 2.10: Left: E-Chalk tool dialog. Below the two action button rows are the radio buttons for selecting a chalk color or the eraser tool. The three sliders at the bottom control the size of the chalk tool, the eraser tool, and typed text, respectively. Right: Dialog for selecting an image to add to the board content.

Drag Handles

The virtual length of the board is not limited. In the setup, the board can be configured to feature a vertical scrollbar. However, the board interface provides another possibility for scrolling which resembles the interaction style of a traditional blackboard more closely. At the top and the bottom of the screen, two *drag handles* are part of the board. When the instructor starts a drag action at one of these handles, he or she moves the board content up or down, like pulling a blackboard up or down. The mouse pointer is shown as a hand icon during dragging and when it is above the drag handle, see Figure 2.11.

Images and Applets

The system allows the user to paste images from local hard drives or from the Internet. Pressing the *add image* button, a dialog displays the image bookmarks with titles and thumbnail icons, see Figure 2.10. The user can pick one of the bookmarks, or type in an image URL, or select an image file with a file dialog from a local disk. The file dialog applies a file filter to show only the supported file formats, GIF and JPEG.[20]

When the user has selected an image, the loading process starts. This process may take a noticeable time, as remote image locations are allowed. If it should take more than half a second, a progress dialog with a cancel button is opened, enabling the user to abort lengthy loading actions.[21] Except for the cancel

[20]PNG images can also be handled by the server software. However, in the default setup their usage is disabled, because the PNG format is not guaranteed to be displayable in remote E-Chalk clients. The client relies on the Java decoding of images, and Java versions before 1.3 lack PNG support in some browsers.

[21]The same approach is taken for other types of resources that can be added to the board, like CGI calls or requests for mathematical computations.

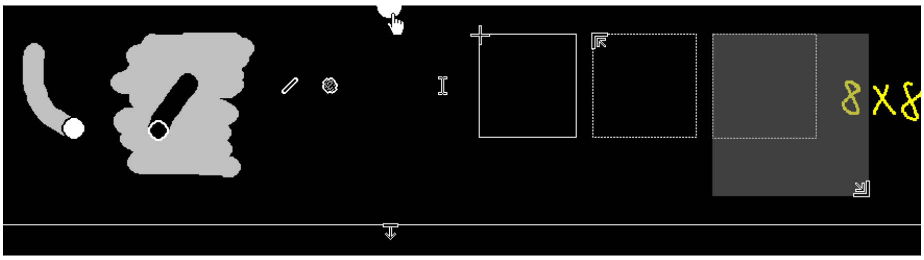

Figure 2.11: Top row, left to right: Drawing-tool mouse pointer, eraser-tool pointer, pointers for tools of big sizes, the hand pointer over the top drag handle, placement of an image or Applet, positioning a chalklet, and dragging the chalklet area bigger, with currently selected area painted semi-transparent over an existing board drawing. At bottom: Positioning a macro.

option, the board does not accept any interaction during the loading process. When the image data have arrived, the user is expected to place the image on the board. The mouse pointer changes to a cross-hair and the bounding rectangle of the image is attached, see Figure 2.11. The image's upper left position is set by a mouse click. Once put on the board, it can be regularly painted with board drawings, letting the instructor annotate the image.

Another button provides the same functionality for adding Applets to the board content. Their URLs reference HTML pages with embedded Applets. When the selected page contains more than one Applet, all of them have to be positioned by the user. Like in a Java-enabled browser, the board then executes the Applets, which can be interacted with regularly[22], see Figure 2.12. Unlike the images, one cannot draw on the Applets with the board tools, because all mouse events on the Applet screen areas are consumed by the Applets.

Typing Text

Another board option is to add keyboard typing to the board content. When the associated button is pressed, one can select the start position for the text input. The positioning to be performed is signalled by changing the mouse pointer to a text cursor, see Figure 2.11. Text typing is then added to the board until the text input is closed by hitting the `[Enter]` key or by adding other contents to the board (a drawing, an image, an Applet, a new key typing, etc.). If the text is too long to fit into a single line, dynamic line breaking is provided automatically. Explicit line breaks can be added by `[Shift]-[Enter]`. Editing by backspace and delete is supported, as well as moving the key cursor in the typed text with the left and right arrow key. Text inputs are stored in a text history and the history entries can be accessed by the up and down arrows.

Keyboard input added to the board content just for writing text are of little value, but they can be used as requests to a connected computer algebra system or to CGIs, as described below.

[22]The replay of Applets actions poses a number of principal problems. For this reason, handling Applets in E-Chalk remained at a very experimental stage. See Section 4.9 for a detailed discussion of the problems.

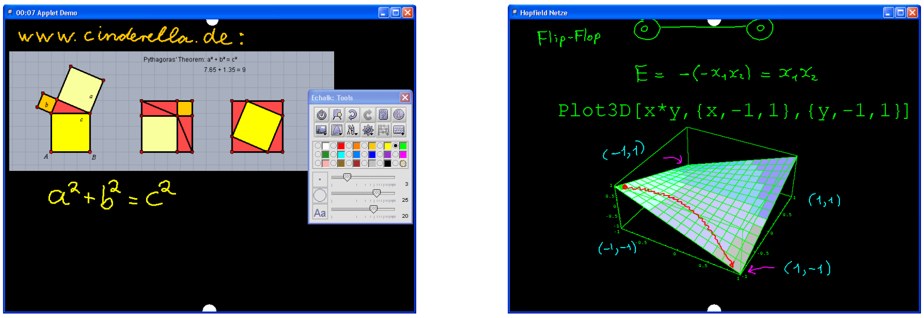

Figure 2.12: Left: An Applet added to the board. Right: Snapshot from a lecture using a plot from a computer algebra system, already annotated by the lecturer.

Requests to Computer Algebra Systems

Mathematical requests can be processed by the board using an interface to computer algebra systems, such as Mathematica by Wolfram Research [Wol03] [108] or Maple by Waterloo Maple [MGH+97] [105].[23] When such a system is installed, the corresponding button allows to type a request, exactly like normal text typings are done. On closing the text input with the enter key, the request is sent to the algebra system and evaluated. If the system returns a graphic, it can be pasted to the board in the same interaction style that other images are positioned. Otherwise, any textual result is typed below the request string. See Figure 2.12 for an example of a plotting request evaluated with Mathematica.

Requests to CGIs

The board also allows to make calls to Web services as bookmarked[24] CGI scripts. When a CGI bookmark is picked from the CGI bookmarks, the user positions a text cursor on the board. For each of the CGI parameters defined in the bookmark, the user is prompted for a value.[25] The CGI is then called and a text or image result is added to the board like for requests to computer algebra systems. See Figure 2.13 for examples.

Mathematical Handwriting Recognition

As mentioned before, keyboard typing in the pen based E-Chalk interface are an interruption of the workflow. Keyboard inputs should be replaced by pen inputs when working in the board metaphor. For this reason, a mathematical handwriting recognition was included in the E-Chalk board right from the very beginning as an interface to the computer algebra systems, see [RKRF01a,RKRF01b].

[23] An interface for MuPAD by SciFace [65] [66] is under development. A built-in system for calculator-type requests and plotting of simple one- and two-dimensional functions has also been written.

The current implementations for Maple and Mathematica require a locally installed algebra system, but it would be easily possible to use a remote server.

[24] See Section 2.4.1 for bookmarking. CGI can only be accessed by bookmarks, as the CGI parameters to be sent over must be declared in advance.

[25] Prompting texts are set in the CGI bookmark editor.

Figure 2.13: Left: CGI request (parameter prompt text "a word:", user input "Kreide") and answer from a CGI script using the *Leo* [52] server. Center: CGI request (prompt "12 digits:") to a CGI returning a EAN code graphic and its result. Right: CGI request for a CGI with several parameters, with the third parameter not yet entered by the user.

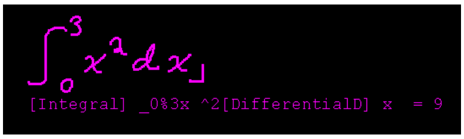

Figure 2.14: The mathematical handwriting recognition employed on the board. The recognized term is printed along with the result from the computer algebra system to give feedback on the recognition result.

While the early version could only handle simple arithmetic expressions, it has now moved to more complex mathematical formulas [FKRT03]. A color can be reserved for input to the recognition module in the setup. Writing a special "tick" symbol marks the end of a handwriting request. The recognized string is sent to the connected computer algebra system (or to a simple built-in calculator, if no algebra system is set up), and the result is added to the board. See Figure 2.14 for an example.

Chalklets

Chalklet are a kind of interactive animations designed for the E-Chalk board. On creation, a chalklet is assigned a rectangular section of the board to operate on. The chalklet then receives what the user draws into that area and may itself draw strokes in the chalklet's area. Chalklets implemented so far include a game of Tic-Tac-Toe, a logic circuits simulator working hand-drawn circuits definitions, and a python interpreter for handwritten programs. See 6.12 for example chalklets.

When a chalklet bookmark is selected, one has to define the chalklet's area on the board. A chalklet defines its minimum area. When a click is used for positioning, the chalklet instance receives this minimum area. Alternatively, the area can be dragged to a bigger rectangle. See Figure 2.11.

Chalklet's areas are allowed to overlap. User-drawn strokes in the intersection of the areas are sent to all the listening chalklets. A chalklet scrolled out of the visible area is deactivated.[26]

[26]This prevents chalklets from changing the board image without the user being able to notice.

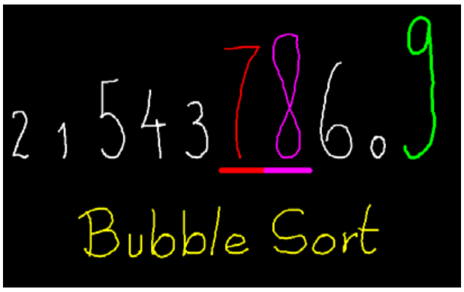

Figure 2.15: A macro for showing the Bubble Sort algorithm as animation. Figure from [Esp04].

Macros

Another kind of bookmarks that can be used are E-Chalk board macros. These are simply board recordings that can be replayed on the board. After selecting a macro bookmark, one has to position the replay by defining the vertical start position, see Figure 2.11. Writing and images from the macro are then added to the board, with the speed they were originally recorded scaled by a factor defined in the macro bookmark. When a new element would be located in a part of the board not visible due to the current vertical scroll offset, the offset is adjusted accordingly, automatically scrolling the board. The macro can be aborted anytime. Any edit to the board other than scrolling stops the macro replay.[27]

Macros can be created like regular lectures with a macro recorder tool[28], or they can be generated synthetically, for example for showing animations.[29] See Figure 2.15 for an example.

2.5 Remote Access

One only needs to open the Web page for the lecture in a browser to view the lecture replay. The replay client is realized as several Java Applets embedded in the Web page. Each of the three possible streams is received by an Applet, the board, audio, and video client. For live transmissions, these Applets open a TCP socket connection to the lecture server.[30] Audio and video clients decode their data streams and output them. For the board, the handling is slightly more complex due to the event-based representation of the board format. In the case of a late connect, it has to load all past events to build up the current board image.

---

[27]A further extension can be the possibility to resume stopped macros and an interactive control of the replay speed.

[28]The macro recorder is described in Section 6.9.

[29]A tool to generate E-Chalk macros for animating algorithms is described in [Esp04], see Section 6.12.2.

[30]Due to the security instructions on Applets, the client Applets can only connect to the Web server they come from. For this reason the E-Chalk main application must run on the Web server in live transmissions. An alternative would be to run a proxy on the Web server, which hands just hands the data over.

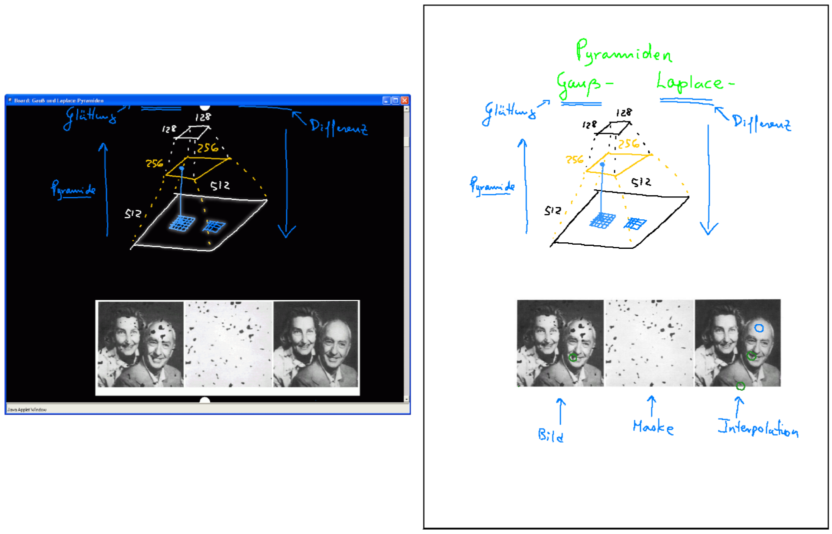

Figure 2.16: Left: An E-Chalk lecture seen in the board Applet. Right: The corresponding PDF page.

The board client allows vertical scrolling by the remote user, enabling the user to look at older board content already scrolled away. This can be done by using a scrollbar featured by the board client or by dragging.[31] Horizontal dragging is also possible when the width of the remote client board is smaller than the board at the server side. This is usually the case when the full board does not fit on the receiving computer's screen, because of its resolution being too low.

For replay from the archive, an additional Applet is opened to provide VCR-like controls, allowing to pause and resume, to fast forward and rewind, and to jump to any point in the recording timeline.

In contrast to the live transmission, this type of access does not connect a remote E-Chalk application. Instead the replay relies on the file-streaming[32] capabilities of the Web server.

In addition to the dynamic board recording, a PDF transcript of the board recording is provided to print out the board content comfortably. See Figure 2.16 for examples of replay and the PDF version.

[31]The dragging is similar to the server side drags using the board's *drag handles*. It just allows to start a drag anywhere on the board surface: drags are not interpreted as drawings on the client board.

[32]The client Applets process the streamed data, i. e. they do not need to load the full recording before starting to play.