

Chapter 2

Automated Lecture Recording

2.1 Distance Education

The word *distance education* was introduced into the English language as part of a new institutionalisation of remote education. In 1969, the Open University [89] was founded in the UK as the first university where no student could attend to a class physically. Soon after, the “Universidad Nacional de Educación a Distancia” (UNED) [90] in Spain was founded (1972), as well as the “FernUniversität in Hagen” in Germany (1974) [79]. Even though the roots of the Internet in the 1960s and 1970s are closely connected to universities, videotapes, the telephone, and in the 1980s also cable and satellite delivery remained the transmission media for distance courses until the end of the 1980s. It took until the 1990s for the Internet to be taken seriously as distance education medium. Then, however, the popularity of distance education started to grow at an unprecedented rate. Today, each of the above-mentioned universities has more students than could be reached by attendance teaching.

Moreover, the ubiquitous availability of connected computers has added a whole new dimension to distance education as regular, i. e., non-distance-education, universities began to see opportunities for a new way of teaching. This approach was given the name *e-learning*. Today, almost every university has its own e-learning project that aims to utilize the opportunities provided by the Internet or other media that allow easy deployment of content to improve education. Among these projects, one can differentiate between pure *distance learning* (a synonym for distance education) approaches and *blended learning*. Blended learning is a hybrid approach that combines e-learning with traditional education methods.

2.2 Technology-Augmented Classroom Teaching

During the 1990s, digital technology also set foot in the classrooms of attendance universities, which began to enhance traditional teaching methods by using computers. Although usage varies from subject to subject and from teacher

to teacher, three approaches can be observed to have reached a predominant position in the field of computer-supported education: Intensive use of slide-show presentations, educational mini applications (e.g., specialized software, dynamic web pages, or Java Applets), and video recording of lectures and their transmission via the Internet.

Slide-show presentations have long since replaced overhead projector slides. The structure of the presentation is entirely planned in advance taking into account all required resources. Visual means like tables, diagrams, images, or even animations can be directly presented to the audience. For distance education use, computer-generated slides may be printed out or put onto web sites, so that students do not have to copy the content for later recalling.

Specialized educational mini-applications are used for presentations in classrooms as well as for individual training of a student who sits in front of his own computer. Pedagogical software like this is particularly common in K12 education because there is a wide range of commercially available programs. Research universities usually prefer to develop their own solutions often targeted to the audience of a single course.

Ever since, distance education has been a way to deploy educational content to more people than would otherwise be reachable using attendance teaching. As a result, many attendance universities have welcomed the upcoming popularity of the Internet and started to think about enhancing classroom teaching by providing students with additional distance teaching lessons. Existing solutions either focus on recording and transmitting a session or using video conferencing tools to establish a bidirectional connection (i.e., a feedback channel) [Knipping, 2005]. Although this approach does not support the teaching process in classroom, recording a video of the entire lecture containing the picture of the board, the lecturer, and an audio track enables students to follow a lecture remotely or to replay previous sessions. A reason often given for combining distance education with classroom teaching is that universities have to cope with an increasing number of students while at the same time universities more often than not have to face drastic cuts in their funding.

With computer technology already being used in classroom teaching, it seems to be but a small step to automate the processes of creating distance learning material directly from classroom teaching. The following sections describe a number of projects and commercial products that were created to automate the creation of distance education material in attendance universities.

2.3 Lecture Recording Without Automation

Often, standard audio/video encoding and broadcasting applications are used to record and transmit lectures over the Internet. The reason for this is primarily their commercial availability and the straightforward handling of state-of-the-art Internet broadcasting software. The downside is that they require manual operation by technical staff as they are not particularly designed for automated recording. Figure 2.1 shows the typical work flow needed for creating web casts with such software. RealNetworks, Inc. write into the “Helix Universal Server Administration Guide” [43]:

Encoding a media clip or broadcast is the last step of a process that involves capturing, digitizing, editing, and optimizing audio

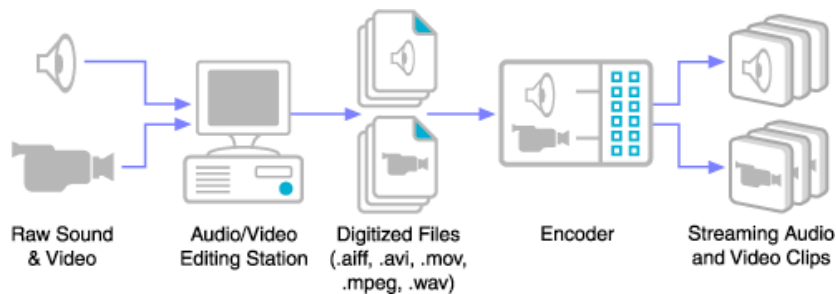


Figure 2.1: Typical work flow for streaming media with commercially available web casting software. Picture taken from the Helix Universal Server Administration Guide [43].

or video data. A streaming media author uses various production tools to accomplish these jobs. These tools typically include video cameras, microphones, recording media such as tapes or CDs, mixing hardware, and audio and video editing software. You can use any tools you want to capture and edit audio and video input. You just need to ensure that your tools can save digitized files in formats that your encoding tools can accept.

However, most of the automation projects described in this chapter are built on top of standard commercial *Internet broadcasting systems*. It is therefore useful to take a more detailed look at the three most important ones, namely the *Windows Media Platform* [34] by Microsoft, Inc., the products from RealNetworks, Inc. [42], and *QuickTime* [8] by Apple, Inc. When looking at the smaller details, there are differences between the three systems. However, these are not important enough to merit discussion of each of them individually. Instead, this section will discuss them all at once. The main scope of commercial encoders is the transmission of audio and video data via the Internet and their digital archiving in files. Each of the systems consists of a three-layer architecture which contains an encoder part, a server part, and a client part. Figure 2.2 illustrates a typical configuration.

The encoder captures live audio and video content and delegates compression to a codec provided by the operating system. Pre-recorded content is handled by using the codec as a converter. Current encoders feature flexible encoding modes with constant and variable bit rates. In addition to stream-capturing devices (such as sound cards, video cards, or FireWire interfaces), encoders are also able to read still images and capture screen shots. Most encoders can capture the entire desktop screen, individual windows, or a region and broadcast or encode it to files. Encoders normally provide a user interface to control everything necessary for live event production, such as pre-defining play lists and switching between live and pre-recorded sources. Several encoders also support controlling conventional hardware devices attached to the computer: Playlists can also include commands like rewind, play, or pause to be sent to digital video cameras and video tape recorders. Mostly, time-code data is also captured from the original source for frame-accurate seeking. Several encoders are also able to integrate presentation slides, for example Microsoft Producer also encodes Microsoft PowerPoint [32] presentations.

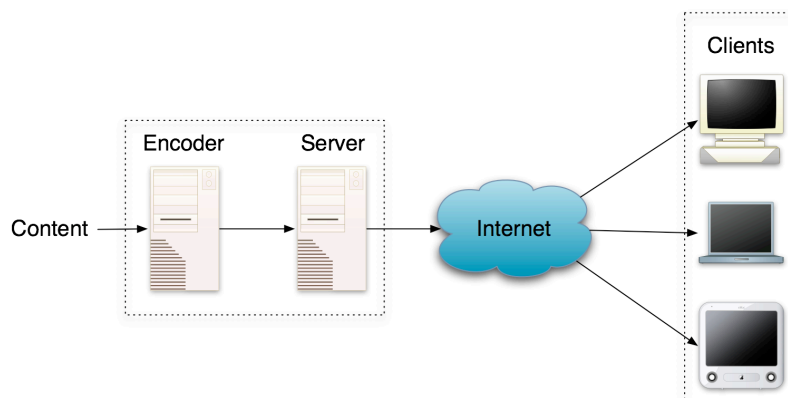


Figure 2.2: The general software architecture of commercial broadcasting servers: The captured data is compressed by encoder software and then broadcast by a server program that may reside on a different computer. Specialized client software receives the stream and plays it back on different hardware platforms.

The server part is able to deliver either a live stream or pre-encoded content over the Internet. For live streaming, a so-called broadcast *publishing point* connects to an encoder and sends the stream to compatible clients. A publishing point is a computer with a properly configured webserver and a running streaming server. Today, most servers can also stream files that were encoded by encoders of different origin. Servers may also get their input stream from the output of another server. This allows for load balancing when many people connect to a certain broadcast at once. Essential runtime parameters such as server load or the number of connections are logged and can be tracked online with the help of a user interface or via SNMP [Case et al., 2002]. The streaming servers can be administrated using different front-ends.

The client program is able to receive a stream sent by a server and to play back files stored on the local harddisk. The Windows Media Player is part of the operating system Windows. Apple's QuickTime Player is part of the operating system Mac OS X. The RealPlayer is part of several operating systems in handheld devices, such as mobile phones. Both the QuickTime Player and the RealPlayer are available on several platforms. Usually, the players install themselves into the webbrowser. They are integrated as plug-ins and are invoked by the webbrowser when a page returns a certain MIME type.

Several universities¹ maintain an infrastructure for the recording and web casting of university lectures or special events. Others also use teleconferencing systems like PolyCom [40] to record and stream lectures or to hold conferences, one example is the University of Indiana [61], which uses a Polycom conferencing system to record lectures before making them available via Apple's PodCasting [6].

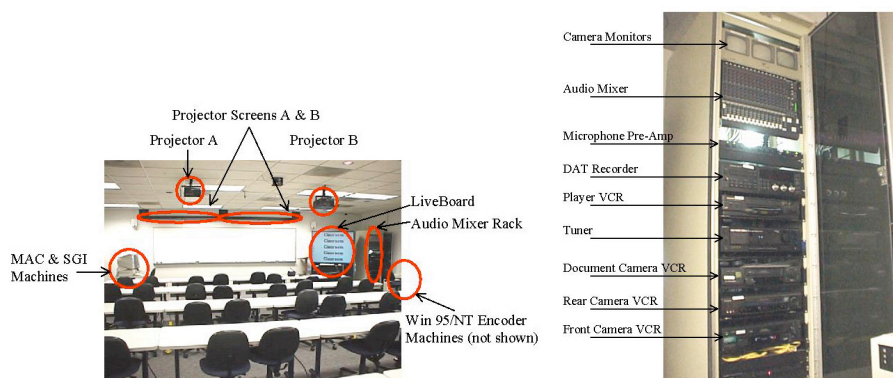


Figure 2.3: The prototype lecture room of the eClass project (left) and the media rack (right). Pictures taken from [74].

2.4 Classroom 2000/eClass

One of the earliest projects to use computer support in the classroom to generate distance learning as a side-effect is the *Classroom 2000* project [Abowd, 1999, Brotherton, 2001], developed at the Georgia Institute of Technology. In 2000, the project was renamed *eClass*. The purpose of the research project, which ended in 2001, was “to study the impact of ubiquitous computing on education”. Classroom 2000 consists of a prototype classroom environment and a software system with the goal to “capture the rich interaction that occurs in a typical university lecture”.

The instructor uses an electronic board system such as *Smartboard* [49], where the computer screen content is projected either from behind or from the front. A pen tracking system is used to simulate mouse movements. Audio recording is done using two dynamic microphones attached to the ceiling of the classroom. However, the instruction manual for Classroom 2000 [74] recommends to use an optional wireless lapel microphone for better audio quality. The audio signal is picked up by a pre-amplifier and an audio mixer rack before it is forwarded to the sound card of the encoding computer. Video recording is done using a front camera for the instructor, a rear camera for the classroom and a document camera to capture non-electronic documents. According to a video downloadable from their project website [74] the estimated minimal costs for an equipment suitable for the system are about \$15,000. Their prototype room cost about \$200,000 including all projectors, computers, and the SmartBoard. The prototype classroom also featured a radio tuner, a VCR, and a DAT player/recorder for the instructor to be able to use non computer-based media during the lecture. Figure 2.3 shows their prototype classroom and the hardware rack.

The recording software (conceptual name *ZenStar*) technically consists of several components that are briefly described in the following. A presentation component called “ZenPad” is used to present pre-specified slides during the lecture. The program also allows for a simple free hand annotation of slides. For the use as whiteboard, the instructor adds an empty slide. The webbrowser used

¹For example: UC Los Angeles [60], Stanford University [50], or UC Berkeley [59].

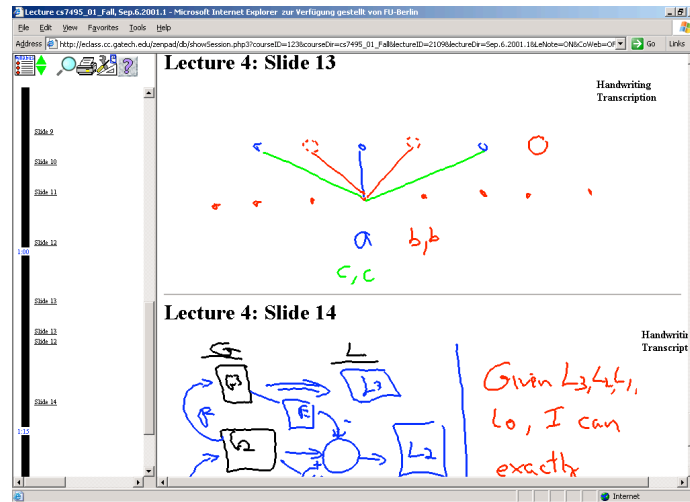


Figure 2.4: A screenshot of a lecture replay with *eClass*.

during the lecture is configured to use a custom proxy server keeping track of every URL visited. A program called “StudPad” allows for student interaction in the classroom. The program distributes the ZenPad content of the presentation computer to any number of student computers. The students can then add private notes. The so-called “ZenStarter” program was used to integrate the different components. The program triggers the “RealProducer” simultaneously with the ZenPad to record optional audio and video of the lecture. After the recording session a program called “StreamWeaver” builds HTML pages including links to the timestamped slide positions that enable navigation inside the recording. The program converts all presented slides, including the last state of the annotation to GIF images and creates a list of links from the logs of the custom proxy server.

The created lecture can be replayed remotely using a web browser. The system replays audio, a small video, any presented slides with static handwritten annotations, and all weblinks visited during the lecture. RealPlayer is required to listen to the lecture audio and to view the lecture video. Figure 2.4 shows an example lecture recorded with *eClass*.

2.5 LecCorder

LecCorder [26] is an early commercial product developed and distributed by CollabWorx, Inc. It is “a lecture capture and publishing system designed for corporate training managers, professional trainers, consultants, and presenters, as well as for academic use”. It is available as an integrated system in either a desktop or a portable version or as a stand-alone software package. The *LecCorder* software system is a Windows-based program that relies on special *MPEG-1* [ISO/IEC JTC1, 1993] encoder hardware². CollabWorx gives a detailed specification of the recommended camera and microphone configuration

²Namely the MovieMaker Express card by OptiBase, Inc.

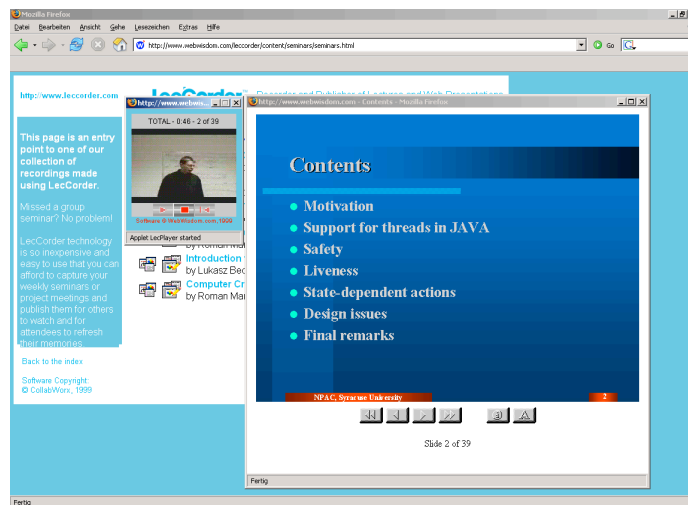


Figure 2.5: A slide presentation recorded with LecCorder.

to achieve proper encoding quality. The integrated LecCorder systems include applicable microphones and cameras, as well as camera lighting. LecCorder uses slides in the form of HTML pages, for example exported by PowerPoint. The presenter uses the “LecCorder system setup tool” to configure several settings such as the output directory for the exported lecture, the input directory of the prior HTML exported PowerPoint slides, and some meta-data. The lecture slides are then copied to a folder that is accessible from the web via an HTTP Server, together with several scripts. During the presentation, the lecturer uses a web browser to show the slides to the audience while the LecCorder software captures audio and video. After the presentation, the audio and video is compressed using the hardware MPEG-1 encoder. The presentation can be replayed using a Java and Javascript-enabled webbrowser. The video is shown at about four frames per second. The bandwidth required for smooth replay is about 64 kbit per second (a single ISDN channel). In addition to the client shown in Figure 2.5 there is also a replay variant where the remote viewer is able to add private annotations to each slide. The personalized slides can be saved to local files.

2.6 Authoring on the Fly

The *Authoring on the Fly (AOF)* project [Bacher et al., 1997, Hürst and Müller, 2001] initiated by Thomas Ottmann at the University of Freiburg was developed to “merge the three activities presentation recording, teleteaching and production of multimedia courseware into one system” [70]. Mostly, the system is used for the annotation of slide presentations. Electronic whiteboards or digitizer tablets are used as input devices. Using a microphone and a camera, the voice and an additional small video of the instructor can be recorded and/or transmitted. AOF uses different software for live transmissions than for archiving lectures. Live transmission is mainly done using standard Unix-based *MBone* [85] tools. Originally, it used the programs “vic” for video transmission, “vat” for

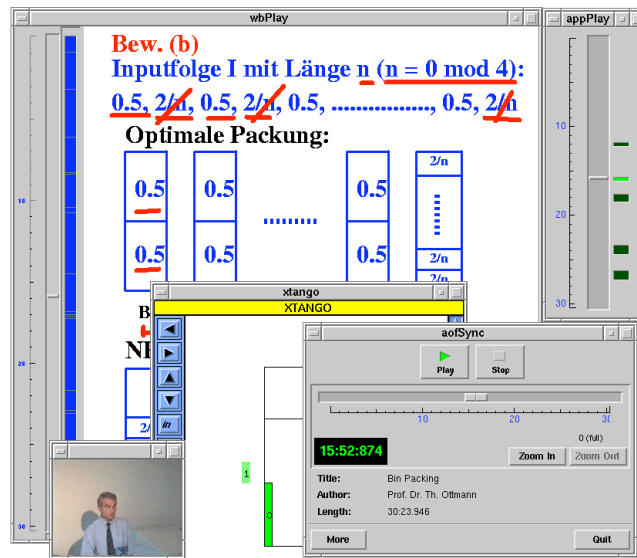


Figure 2.6: A screenshot of a lecture replay with AOFsync. Picture taken from [70].

audio transmissions, and “wb” for the transmission of whiteboard content and slides. The tools are all part of the *MBone tool set* [86] and were originally developed for conferencing. The programs were replaced by “AOFwb” and later by a program called “Media Lecture Board (mlb)” running on Windows. The tool handles board strokes as vector graphics and can import slides, images, and also video files in different formats. Audio is transmitted and archived without compression using a sampling rate of either 8 or 16 kHz (8 bit mono).

Live lectures are received using a special client program called “AOFrec”. The client program is a small standalone Unix program that is not coupled to a web browser. Closing “mlb” results in the creation of HTML pages that enable asynchronous replay of the lecture using a client called “AOFsync”, of which there is also a Java variant called “AOFJSync”. Figure 2.6 shows a screenshot. Both clients require the download of the entire recorded material to a local harddisk prior to replaying the lecture because the program is not embedded into a web browser either. AOF provides several tools to automate packaging of lectures and to burn CD-ROMs so that lectures can be directly replayed off the disk. There are also tools for indexing lectures with keywords and for the creation of directories over a set of recorded lectures. A number of additional tools allows for post-production of lectures and for direct import of PowerPoint slides into mlb.

2.7 Lecturnity

Lecturnity [25] is a commercial spin-off of the Authoring on the Fly project (see Section 2.6). However, it does not provide live transmission of lectures. Lecturnity integrates into PowerPoint and mainly consists of three programs. “Lecturnity Assistant” is the main recording component. Lecture recording is controlled

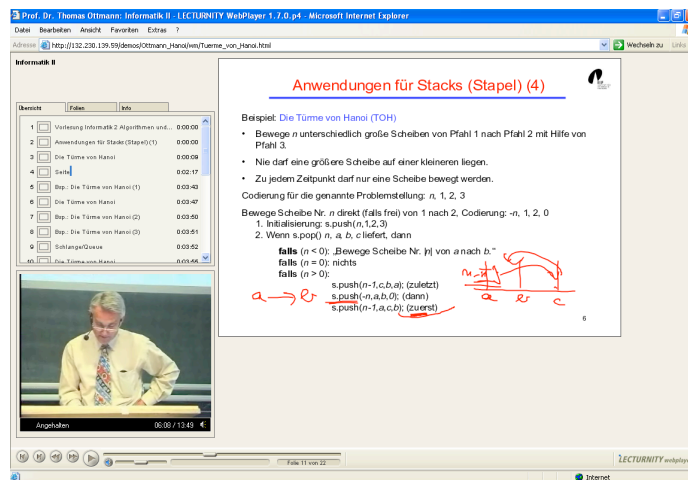


Figure 2.7: A screenshot of a lecture replay with Lecturnity.

by a simple console that provides start/stop/pause functionality. The slides can be annotated using different painting and drawing facilities. Capturing of slide content and annotations is actually done by screen grabbing. The entire desktop screen or only a part of it is captured and recorded in addition with audio and an additional video feed. The video signal is mostly used for recording a small video of the instructor or the classroom environment. As screen grabbing forms the principal recording component, lecturers can also record the use of any other application and are not limited to the use of PowerPoint.

The “Lecturnity Editor” allows for post-production of recorded lectures. It provides the conventional audio and video editing and filtering facilities and allows to cut, copy, and paste entire lecture parts. A program called “Lecturnity Publisher” is provided to package a recorded lecture for distribution on CD or DVD, as well as for publication in the Internet. The packaged lectures are embedded into HTML and can be played back with RealPlayer, Windows Media Player, or Macromedia Flash. Lectures can be navigated using a slider or by selecting a specific slide from a thumbnail index. Figure 2.7 shows a screenshot from an example lecture replay created with Lecturnity.

2.8 Lectopia/iLecture

The *iLecture* system [William and Fardon, 2005], recently renamed *Lectopia* in several countries, is a commercially available system initially developed at the University of Western Australia. In 1998, the first version of Lectopia was developed for the purpose of replacing a small lecture tape library service that supported the automated capture and processing of lectures from audio tape to help out part-time arts students who had difficulties attending lectures. According to [82], “the main driving force behind the development of the *iLecture* System was the desire to make lectures and associated material available to all students with Internet access, at the time of their choice”. The system assumes a pre-configured classroom, with at least a camera and a microphone. It may

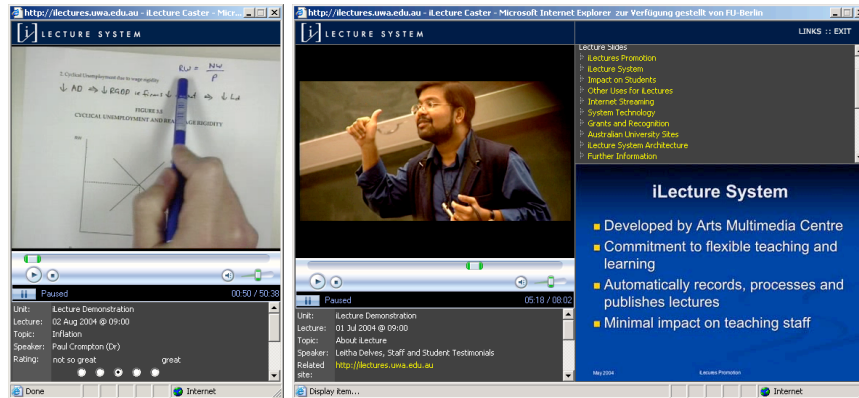


Figure 2.8: Screenshot of a lecture replay with the iLecture system using a document camera (left) and using a camera for the instructor and additional unsynchronized slides (right). The bandwidth required for the examples is 1 Mbit/s.

also be used for screen capturing. For the recording of blackboard and paper content, the authors recommend the use of a *document camera*. This is a regular camera, entirely focused on the chalkboard or a sheet of paper. Figure 2.8 shows a sample lecture produced with such a camera and a replay of a lecture recorded with a typical setup.

The core of the software is the “iLecture Administration Tool”. This is a web-based information system that enables administrators to schedule lecture recordings at certain times. Once a recording has been scheduled, Lectopia takes care of capturing, publishing, and notification of availability of the recording by using various off-the-shelf components, such as RealProducer, Windows Media Encoder, and QuickTime. The actual lecture recording is started by microphone activity and stopped when the time schedule ends (regardless of microphone activity). The lecture can be broadcast live and/or stored in a database that is accessed by remote viewers through the content management system *Lasso* [37] or the learning management system *WebCT* [69]. Slides have to be uploaded separately and are not synchronized with the stream, i. e., the user must click on a link to open each slide manually while looking at the recorded video of the lecture. Lectopia also supports the automatic conversion of lectures to formats that are optimized for small devices, such as PDAs or mobile phones. The Lectopia Server also provides an API that allows external information systems to search, retrieve, and change information in the system programmatically. The API is based on the *Simple Object Access Protocol (SOAP)* [66].

2.9 Camtasia

Camtasia [12] by TechSmith cooperation is a commercial screen grabbing tool. Although not explicitly developed for lecture recording, it is regularly used [Burcham, 2003] for this purpose by several universities.³ The reason for this is that Camtasia’s functionality is very similar to Lecturnity (see Section 2.7). The

³See for example lecture recordings at RWTH Aachen, University of Freiburg, or Stanford University.

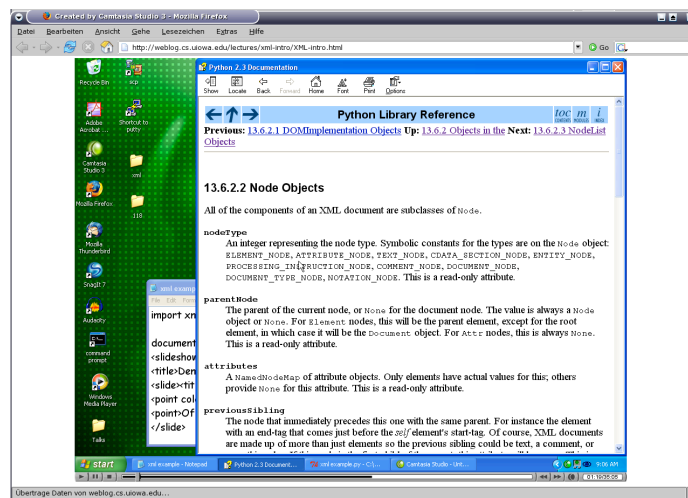


Figure 2.9: A screenshot of a university tutorial recorded with Camtasia played back in a web browser.

software allows screen grabbing at different frame rates synchronized with audio and an optional video stream, integrated picture-in-picture. The result is a video file that can be encoded in different standard formats, like Windows Media AVI, QuickTime MOV, RealMedia, Macromedia Flash or animated GIF (without audio). Camtasia provides editing facilities and tools that ease the production of video DVDs. Figure 2.9 shows a replay of a university tutorial recorded with Camtasia from [62].

2.10 tele-TASK

Tele-TASK (*tele-Teaching Anywhere Solution Kit*) [Ma et al., 2003, Meinel et al., 2005] is a university project directed by Christoph Meinel at the Hasso-Plattner-Institut at Potsdam. The main goal of the project is to minimize the efforts required for lecture recording by integrating the required hard- and software into one device, the so-called *tCube*. Similar to other systems presented in this chapter, the Tele-TASK system grabs the screen content combined with audio and an additional video feed of the presenter. The systems assumes a presentation computer, a microphone, and an optional video camera, all connected to the *tCube*. The project homepage [41] recommends the use of an electronic blackboard to project the presentations onto “in order to be able to add handwritten remarks”. The software of the *tCube* includes a RealEncoder, which encodes the audio and video stream into RealVideo format. The screen content is captured at one frame per two seconds. The system is able to transmit the recordings live using an additional computer acting as server. However, when transmitting live, the screen content is not compressed and requires a high-bandwidth connection (at least 1 Mbit/s). After the recording session has been terminated, a post-processing step compresses the screen content using differential frames in PNG format. The slides are synchronized with video and audio using *SMIL* [67]. The compressed recordings can be viewed with a modem (56 kbit/s). Figure 2.10 shows a lecture replay recorded with the *tCube*.

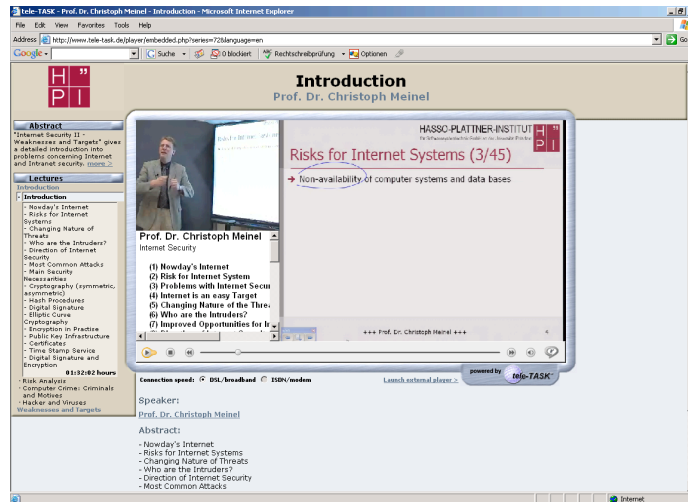


Figure 2.10: A lecture recorded with tele-TASK.

2.11 Classroom Presenter

Conference XP is a project conducted by Microsoft Research. The initiative is to explore “how to make wireless classrooms, collaboration, and distance learning a compelling, rich experience by assuming the availability of emerging and enabling technologies, such as high-bandwidth networks, wireless devices, Tablet PCs” [33]. *Conference XP* is a software development kit (SDK) based on Windows XP that is to serve as a research bed for exploring the creation of distributed applications using Tablet PCs and wireless networks. *Conference XP* is based on a four-layer architecture: an application layer, a capability layer, a conference API, and a network transport layer. Several complete applications are already included in the SDK, these are mainly configuration wizards and non-specialized, extensible tools, such as the *Conference XP* client. The client is a generic peer-to-peer video and audio conferencing application. The capability layer provides graphical user interface components that can be reused by applications using the SDK. Both the *Conference XP* applications and the capabilities layer use the *conference API*. The conference API provides standard protocols to transfer documents in different formats to enable remote interoperability of applications. It also allows the vector-based transmission of ink strokes. Microsoft research claims that the API conforms to the IMS/SCORM interchange specification [2]. In addition to interoperability, the API also eases access to the DirectShow and Windows Media API (see 2.3) that provide the operating system’s audio and video features in Windows. The network transport layer provides access to low-level network transport protocols and is the bottom-most layer in Windows XP. Microsoft Research has teamed up with research organizations and universities in several research projects based on this SDK. One of these projects is *Classroom Presenter* [Anderson et al., 2006]. It aims to improve “computer-based presentation systems [that] severely limit flexibility in delivery, hindering instructors’ extemporaneous adaptation of their presentations to match their audiences. One major limitation of computer-based

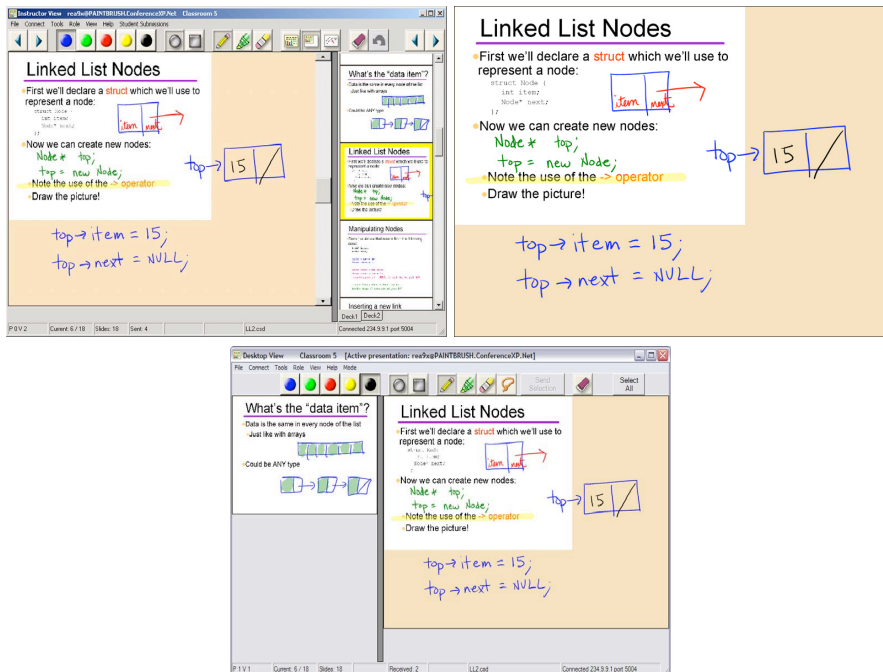


Figure 2.11: Screenshots of Classroom Presenter: instructor view (left), classroom presentation (right), student view (below). Students are able to make personal annotations, these can also be sent to the instructor computer. Pictures taken from [63].

systems is lack of support for high-quality handwriting over slides, as with overhead projectors and other manual presentation systems” [44]. The system is to combine “the advantages of existing computer-based and manual presentation systems and build on these systems, introducing novel affordances”. Technically, the system is an extension to MS PowerPoint that allows writing on top of slides. When lecturing using Classroom Presenter, the instructor writes on top of images of the slides which are directly projected from a Tablet PC and shown on networked remote computers. Using their own networked Tablet PCs, students can also write on the slides and send these notes to the instructor PC. Figure 2.11 shows sample screenshots of the system.

2.12 A Minimalistic Automated Lecture Recording System

In order to see what can be done using only off-the-shelf components, the *FU PowerPoint Recorder* was developed by Damian Schmidt. The task was to create a PowerPoint plug-in that allows the recording of video and audio feeds synchronized with the slides. The required user interaction overhead was to be minimized to one mouse click. The recordings had to be replayable in a web browser. The system consists of about 2,000 lines of Visual Basic code and embeds itself into PowerPoint as a macro. When the user enters PowerPoint’s

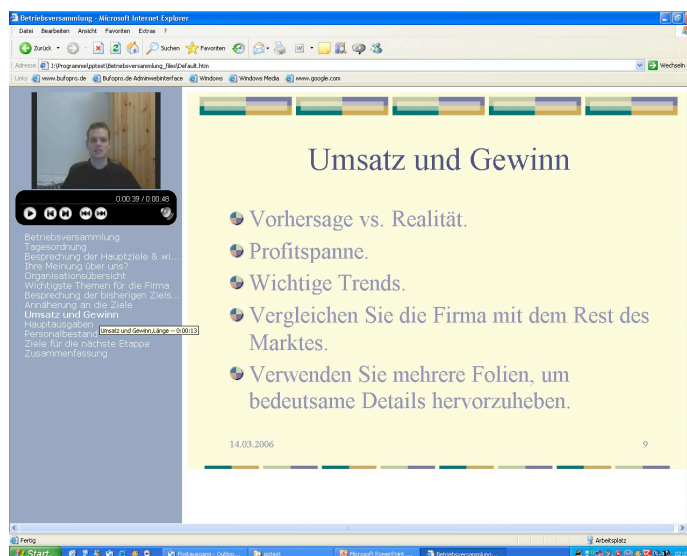


Figure 2.12: A screenshots of the replay of a presentation automatically generated by the FU PowerPoint Recorder.

presentation mode, an additional dialog screen is shown. The user can press the “cancel” button to proceed a normal presentation or “ok” to record the presentation. The script initiates the capturing of audio and/or video, provided the operating system detects a sound card and/or a camera. Operating system codecs are used to compress the captured streams. Slide transitions are reported to the script by PowerPoint itself. After the presentation has finished, the script generates a batch program that controls Microsoft Producer. Microsoft Producer converts the recorded data into Windows Media Video format and generates suitable web pages for replay. Figure 2.12 shows a screenshot of a resulting web presentation. It can be replayed using a web browser and Windows Media Player. The entire process does not require any user interaction. The required bandwidth depends on the codecs that are used. A low-quality version of a slide presentation can be replayed with audio and additional video using a 64 kbit connection. The experiment shows that a slide-show recorder can be written in a few thousand lines of script code. Animations can be replayed and the system is able to playback handwritten annotations on the slides. Of course, this “poor man’s slide recorder” has several drawbacks. It is a proprietary solution that only works for Microsoft PowerPoint in combination with Microsoft Producer. There is no means to integrate any other application. The recorded lectures can only be played back using versions of Internet Explorer and Windows Media Player, which are only available on Windows XP. At least a DSL or cable connection is required to receive such a lecture remotely in adequate quality. Live transmissions are not possible. The encoding process after the lecture takes about 1.5 times the lecture length. This makes it impossible to record two lectures in immediate succession on the same computer.

2.13 Other Systems

There are numerous other systems that cannot be explained in detail here. One of them is *DyKnow*, which is being commercially distributed by DyKnow, Inc. [15]. The system is very similar to Classroom Presenter with the exception that it is not an extension to MS PowerPoint but brings its own core program, which looks like a standard vector-drawing application. VCPlayer by [Sheng et al., 2005] is also very similar to Classroom Presenter but it allows to embed a small video of the instructor and is not specialized for Tablet PCs. *Tegrity* by Tegrity, Inc. [55] is very similar to iLectures. The main difference is that it allows students to add their own handwritten notes to a recorded lecture and the recording system is sold as all-in-one hardware solution. The *TeleTeachingTool (TTT)* [Ziewer and Seidl, 2004] developed at the University of Trier is very similar to AOF. A Java-based client application plays back annotated slides together with the voice and a small video of the instructor. The *Virtual Director* [Machnicki and Rowe, 2002], developed at UC Berkeley, helps automating the process of producing Internet web-casts. It saves man-power by enabling several webcasts to be run by a single technician. The system selects which streams to broadcast and controls other equipment such as moving cameras to track the speaker. The automation of multimedia productions is also researched in [Davis, 2003a, Davis, 2003b]. A “proactive capture device” interacts with the user and the environment in order to create annotations and footage for every sequence of pictures captured. The picture sequences are then collected into a pool and can serve as building blocks for different movies. The underlying motivation is very similar, namely, making multimedia content creation easier for everybody. The work, however, aims more at automating video direction in a cinematographic sense.

2.14 Conclusion

Internet broadcasting systems were created to perform traditional radio and TV broadcasting using the Internet. They are designed to fit into the work flow of a commercial radio or TV broadcasting station. Using them in a university for lecture broadcasting actually requires technical personnel for operation. Because personnel is seldom available, several university projects try to automate lecture recording and transmission by building automation mechanisms on top of these systems. The biggest downside of this approach is that all content has to be transmitted and archived as video or audio track. Slides as well as desktop activity is often captured by a screen grabber and then encoded as video. Often, this results in poorly encoded material that needs much more bandwidth than necessary. Even when proper screen codecs are being used that avoid some encoding problems, simply starting several monolithic stream encoders simultaneously is not an optimal solution for recording and archiving lectures. The multimedia content creation work flow as assumed by Internet broadcasting systems (illustrated in Figure 2.1) is short-cut: There is no audio or video technician supervising the recording of the lecture and there is no editing step between encoding and transmission. This results in quality degradation which is especially noticeable in the audio track (Chapter 7 will discuss this topic in detail). Mapping any content to video format results in a huge semantic gap. It

does not allow for domain-specific encoding and makes post-editing and partial reuse of archived content cumbersome. Using video formats for replaying text or handwritings that could otherwise be played back in vector-graphic formats is also bandwidth inefficient⁴. When using traditional Internet broadcasting systems, the computer is only used for transmission, storage, and replay of audio and video. Without manual editing, the integration of different content streams is reduced to a simple parallel playback in different windows on the desktop. Chapter 9 will report on several problems that result from this lack of integration.

⁴An interesting early comment on this topic in the web can be found at [20].