

# Recording Instructional Presentations on Your PC

Wolfgang Schreiner  
Engineering for Computer-based Learning  
University of Applied Sciences at Hagenberg, Austria

March 19, 2003

## **Abstract**

We describe a low-cost technical scenario of digitally recording an instructional presentation. The required equipment is minimal; the necessary work can be performed by an individual instructor. With some practice, the scenario does not involve much more work than giving a presentation in class.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>An Instructional Presentation</b>	<b>6</b>
<b>3</b>	<b>Technical Requirements</b>	<b>6</b>
3.1	The Lecturer . . . . .	6
3.2	The Student . . . . .	9
<b>4</b>	<b>Setup</b>	<b>9</b>
<b>5</b>	<b>Recording</b>	<b>11</b>
<b>6</b>	<b>Post-Processing</b>	<b>13</b>
<b>7</b>	<b>Web Transport</b>	<b>16</b>
7.1	Bandwidth Requirements . . . . .	16
7.2	Web Server . . . . .	16
7.3	JavaScript Controls . . . . .	16
7.4	RealMedia File Wrapper . . . . .	17
7.5	Presentation File . . . . .	17
7.6	Index File . . . . .	18
7.7	Usage Logging . . . . .	18
<b>A</b>	<b>Software Links</b>	<b>20</b>
<b>B</b>	<b>Camtasia Recorder Configuration</b>	<b>21</b>
<b>C</b>	<b>Wacom Graphire2 Configuration</b>	<b>22</b>
<b>D</b>	<b>Player Control</b>	<b>23</b>
D.1	flashRealNavigation.js . . . . .	23
D.2	FlashNavigation.js . . . . .	25

<b>E</b>	<b>Presentation File</b>	<b>26</b>
E.1	Flash Presentation with RealMedia Stream . . . . .	26
E.2	Flash Presentation . . . . .	28
<b>F</b>	<b>Dummy Authentication</b>	<b>29</b>

# 1 Introduction

Dynamic electronic contents (animation, video, software) offer new didactic possibilities compared to static ones (lecture notes, slides). They provide additional media (sound, voice, moving pictures) and, in the case of software, automatic feedback. However, their production is a time-consuming, expensive, and difficult task which only pays off if the content can be reused in multiple courses which are attended by a large number of learners. These requirements are not met by academic scenarios where a lecturer teaches a moderate number of students in a specialized and rapidly changing field (as is the case for technical subjects) with little or no standardization of curricula (as is the case in much of Europe).

For these reasons, it is hardly possible to find of-the-shelf content (be it electronic or not) that meets the requirements of a particular course. Therefore the typical academic lecturer continuously produces and revises her own content objects from textbooks, magazines, and Web resources, often borrowing material from other lecturers. The results are sets of lecture notes and presentation slides that are far from perfect but at least address the requirements of the moment; a good deal of it can be reused in a repetition course but some part of it will have to be changed soon. The time economy of content production is therefore of crucial importance to every lecturer and the production of dynamic electronic contents is often deemed too labor-intensive.

There are two principal (non-exclusive) strategies to overcome the problem:

1. The formation of content producing academic communities where each partner contributes by a small content object to the overall good;
2. the devise of content production schemes that are cheap enough to be used by a single lecturer and whose results are good enough in practice.

In this paper, we address the second, i.e. the “cheap enough/good enough”, strategy by a scenario where a lecturer may record an instructional presentation on her PC.

The idea of such presentation recording scenarios is not new. The Audio-graph Toolset [1] records audio and sequences of multimedia elements (slides, pictures) that can be streamed over low-bandwidth modems. The commercial system Tegrity [7] records a presentation by simultaneously capturing multiple data sources (video camera, whiteboard drawing, Powerpoint presentation, screen content); the recording can be streamed to the Web by a

special server. Similarly the commercial LecCorder system [3] is able to produce an audio/video stream synchronized with presentation slides delivered as HTML pages; Java applets display the results in a Web browser. [2] and [6] discuss the use of lecture recording for campus-based universities and give an overview on various related technologies.

This paper adds to above developments a simple office desk (not lecture hall) scenario with a single person performing all the necessary steps (recording, editing, publishing). The recording is finally encoded as a Macromedia Flash movie [4] for which a player is claimed to be installed in more than 500 million browsers [5] and that can be delivered by any Web server. The bandwidth requirement is about 100 kBit (including the video), i.e., we do not support the use of modems. Rather we demand (e.g. ADSL-based) high-bandwidth connections which have started to replace modems in private homes at such a rate that we can expect them to be standard in a couple of years.

In more detail, we use for our recording the following media channels:

1. (a portion of) the PC's screen area which embeds visual presentations based on slides, electronic whiteboards, and software demonstrations;
2. the voice of the lecturer which actually transports most of the information of the presentation;
3. (optionally) a video of the lecturer which does not transport much information but enhances the learning experience psychologically.

The first channel must be media-encoded without loss (any imperfection in the screen recording is experienced as detracting). The second channel can be encoded at some loss with better than phone-like quality (small imperfections in voice recordings are easily compensated by our brains). The third channel can be highly compressed with corresponding loss of quality (the presenter's video just serves as a companion to the presentation).

We claim that by such a recording the factual information can be transported in a way that is quite close to that of a typical lecture hall setting. Moreover, a student may play a recording via the Internet at any time or place he chooses and he may play it as often as he chooses; he may pause and restart the recording and may move it forward or backward in a VCR-like fashion. The recording can be produced by a single lecturer with some experience in about twice the time that a corresponding lecture takes place (including preparation and post-processing). Most of the remainder of this paper is dedicated to the explanation of this production.

Apparently, a presentation recording does not support any interaction between teacher and student or among students. Therefore, presentation recordings can be only one component in a blended learning scenario that must also involve other components (technologies and face-to-face meetings). The design of such a scenario is beyond the scope of this paper.

## 2 An Instructional Presentation

Figure 1 displays a sample presentation recorded by the means described in the following sections. The presentation consists of

- a screen area where the lecturer can present slides, draw on a whiteboard, and in general perform any activity he wants;
- an area where a video of the lecturer is shown;
- an area with buttons that can be used by the student to control the presentation in a VCR-like fashion:

**Play** starts the presentation respectively restarts it after a pause.

**Pause** temporarily suspends the presentation.

**Rewind** resets the presentation to the initial state.

**Forward** moves the presentation forward by a chosen amount of time.

**Back** moves the presentation back by a chosen amount of time.

**60s** is the content of an input area where the amount of time can be chosen by which the presentation is moved forward/back.

**482s** is the content of an output area where the current position of the presentation is displayed.

The presentation is displayed in a separate browser window with no controls apart from the window buttons that allow to minimize/close the window.

## 3 Technical Requirements

### 3.1 The Lecturer

For the production process, we have used the following hardware equipment:

Binary Numbers Mozilla

**Character Encodings**

- Text is a sequence of characters.
  - H i , H e a t b e r
  - 72 105 44 32 72 101 97 116 104 101 114 46
- ISO 8859-1 contains 256 — 2<sup>8</sup> characters (Latin 1).
  - First 128 characters coincide with ASCII standard.
- Unicode contains 65,536 = 2<sup>16</sup> — 2 characters.
  - First 256 characters coincide with ISO 8859-1 standard.

The ASCII characters are the same in all encodings.

08.02.022

414 ASCII → 0 127

255 ISO 8859-1 → 0 127

Unicode → 0 65535

2<sup>16</sup>-1

2<sup>8</sup>-1

Play Pause Rewind Forward Back 6C s 42s

3

Figure 1: A Sample Presentation



Figure 2: webcam, Headset and Graphics Tablet

1. A PC with 900 MHz Pentium III, 256 MB memory, 20 GB disk;
2. a 19" screen with 1152x864 pixels resolution;
3. a sound card and speakers;
4. a LabTec Axis 502 headset with microphone;
5. a Philips PCVC740K ToUCam Pro USB webcam;
6. an A6 Wacom Graphire II graphics tablet.

In addition to our standard PC equipment, we had to buy the headset, the webcam, and the graphics tablet at an overall price of 250 Euro or so. The webcam is only required if you want to have the presentation accompanied by the lecturer's video.

Our software equipment consists of (see Appendix A):

1. Camtasia Studio 1.0.0 (for screen recording and post-processing),
2. Philips VRecord (for video recording, part of the webcam software),
3. MS Netmeeting 3.01 (for the white-board),
4. Adobe Acrobat Reader 5.0 (for the slide show),
5. Actual Title Buttons 1.8 (to keep the slide show in front of the white-board).

In addition to our Microsoft Windows environment, we had to buy Camtasia Studio and Actual title Buttons at an overall price of 380 Euro or so.

For the delivery of the recording via the Internet, access to any Web server is required.

### 3.2 The Student

To play the presentation recordings, the student needs access to the following hardware equipment:

1. a PC under Microsoft Windows or Linux,
2. a sound card and speakers/headset,
3. a screen with resolution 1024x768,
4. an Internet connection with at least 100 kbit/s bandwidth.

In principle, the recordings can be also delivered by CDs; then the last requirement can be dropped.

Furthermore, the following software is required (see Appendix A):

1. the Internet Explorer 6 (Windows) or the Mozilla 1.2 (Windows, Linux) Web browser with JavaScript enabled,
2. the Flash Player 6 plugin,
3. (optionally) the RealOne Player plugin.

All necessary software components are freely available.

## 4 Setup

We use the **Camtasia Recorder** to record an instructional presentation as an AVI file; furthermore we use the **Philips VRecord** utility to record the lecturer's video as an AVI file. The necessary steps for the setup are listed below; for further configuration information, see the appendix.

1. **Screen and Audio Recorder.** Start in folder `Programs/Camtasia Studio/Applications` the program `Camtasia Recorder`. If you have already configured the recorder (see Appendix B), then just press the red button: a green rectangle appears on the screen denoting the area to be recorded; the recording is still paused. Position the `Camtasia Recorder` window outside of the recording area.
2. **Whiteboard.** Start `Microsoft Netmeeting` and press the “Whiteboard” button at the bottom row; a whiteboard pops up and you may minimize the `Netmeeting` window. Position the whiteboard such that it fully covers the recording area. Choose the menu option `Tools/Line Width` and select the thinnest line width; select with the right button the `Pen` as the writing device.
3. **Slide Show.** Start the `Adobe Acrobat Reader` with your slide presentation and position it such that it covers about 2/3 of the recorded area (the green boundary itself is not any more part of the recording area). Press the “Stick on Top Button” of the `Actual Title Buttons` tool in the menu bar; the slide will now stay on top while you write on the whiteboard.
4. **webcam Recorder.** Start in folder `Programs/Philips ToUCam Camera` the program `Philips VRecord` and configure it as follows:
  - (a) Select the menu option `Options/Video Properties`, deselect the `Control/Full Auto` option, select `Frame Rate 5` and the modes `Mirror Image` and `Backlight compensation`.
  - (b) Select the menu option `Options/Video Format` and choose as `Output Size` the value `160 x 120`.
  - (c) Select the option `File/Allocate File Space` and choose a `File Size` of `100 MB` (this is the space reserved for the capture file).
  - (d) Select the option `File/Set Capture File` and choose a file location on the local disk (not a network disk located on another machine).
  - (e) Position the window outside of the recording area and select the menu option `Capture/Start Capture`. A confirmation window will pop up, i.e. the recording does not start yet.
5. **Graphics Tablet.** Position your tablet conveniently in front of you and check that the writing area matches the screen recording area. If not, then configure your tablet accordingly (see Appendix C). Write a bit on the whiteboard and then start a new slide (right bottom button).

6. **Sound Check.** Start in folder `Programs/Accessories/Entertainment` the program `Sound Recorder`. Record a small speech and play it to check the recording quality and to adjust the microphone.
7. **Video Check.** Position yourself as for the presentation and adjust the webcam accordingly.

At the end of these steps, the screen has a layout similar to that shown in Figure 3.

## 5 Recording

Make sure that there is sufficient free space on your local disk; for a 30 minutes recording, the Camtasia Recorder produces a 50 MB AVI file (the file space for the webcam recording is allocated by `Philips VRecord` before the recording starts).

Make sure that you are not going to be interrupted (people entering the office, phone ringing, etc).

To start the recording, click with your mouse on the confirmation button of the `Philips VRecord` utility and *simultaneously* press the F10 key. Make sure that the confirmation window has disappeared and that the green rectangle is blinking. The recording has now started.

During the recording, keep the following in mind:

1. Look into the webcam from time to time.
2. Keep the button of the tablet pen on the back such that you do not involuntarily click on it (this can become quite annoying).
3. Do not record much longer than 30 minutes. Apart from you and the audience getting tired, for longer presentations the sound and the screen recording get noticeably out of sync.

To stop the recording, select in the `Philips VRecord` utility the menu option `Capture/Stop Capture` and press in the `Camtasia Player` the button with the black square.

In case you have to interrupt the recording, stop both `Philips VRecord` and the `Camtasia Recorder`. You can still produce a single presentation from

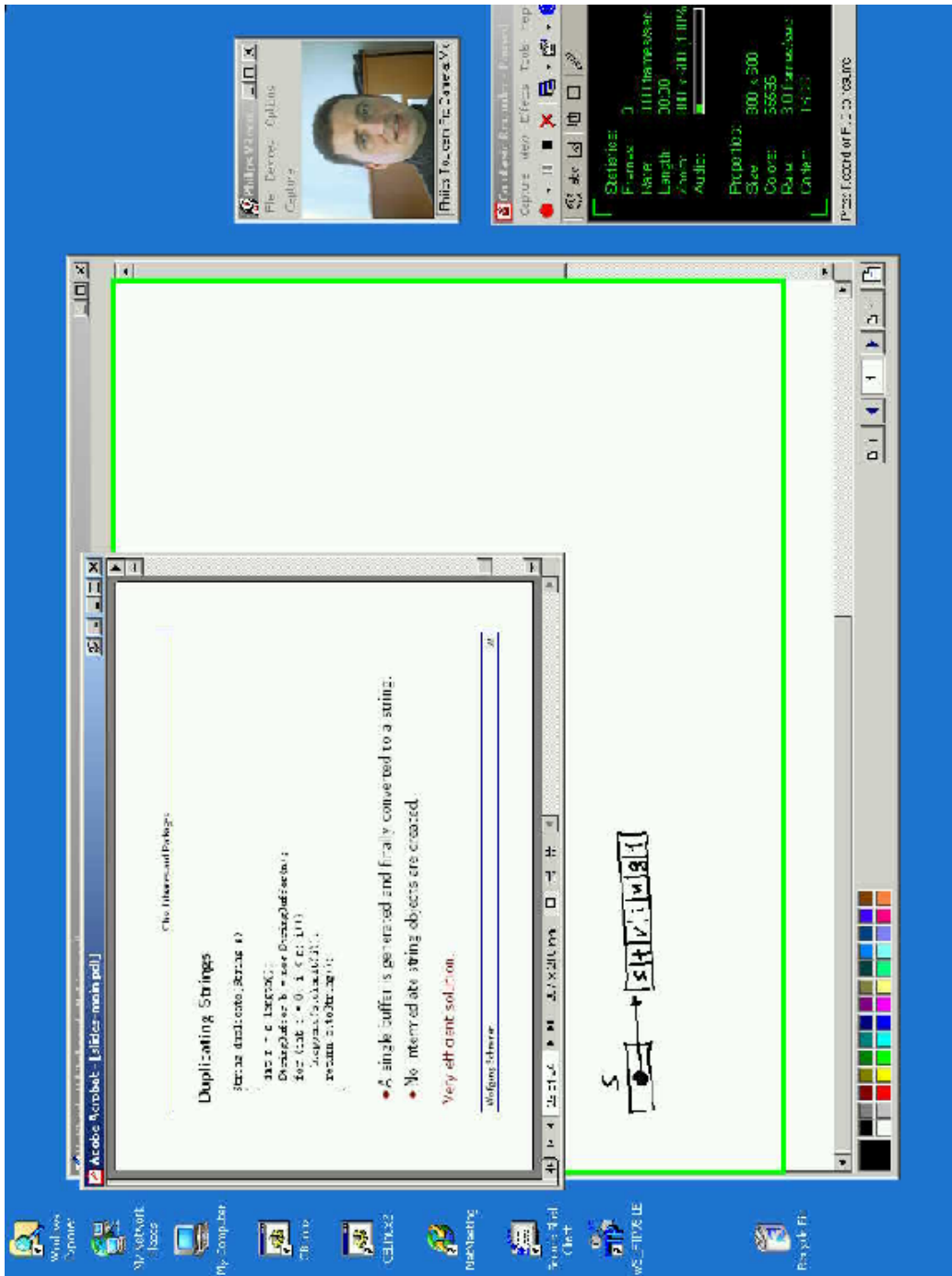


Figure 3: Recording a Presentation

multiple recordings (see the next section); for the moment, write down the names of the recorded files and start a new recording at the point where you were interrupted. Do not forget, however, to choose for the new recording a fresh capture file in **Philips VRecord**, because the old one is overwritten otherwise.

## 6 Post-Processing

The AVI file generated by the **Camtasia Recorder** (carrying the screen and the audio recording) is converted into a Macromedia Flash file. Optionally, the AVI file generated by the **Philips VRecord** utility (carrying the video recording) is converted into a RealMedia stream. For this purpose, we apply the **Camtasia Producer**. The necessary steps are listed below (see Figure 4).

1. Start in folder **Programs/Camtasia Studio/Applications** the program **Camtasia Producer**.
2. **Macromedia Flash Production**
  - (a) Select in the leftmost tab the directory of the the AVI file, then in the medium tab the file itself and move it to Storyboard (the top area). Then select the menu option **File/Produce Movie**.
  - (b) Choose as **Movie File Format** value **Macromedia Flash (SWF) File** and press the **Options** key.
    - i. Select the **Encode Audio** option and choose as **Audio Format** the value **ADPCM**. Select as **Audio Attributes** value **22.050 kHz; 2 Bit; Mono; 22 Kbit/sec**.
    - ii. *Unselect* the option **Show Playback Controls**. Select as the **End Action** the value **Stop** and select the **Pause at Start** option.
  - (c) Press the button **Produce**; the production process will take about 1.5 times as much as the recording length. The result is a **.swf** file and a **.html** wrapper file; the movie is immediately played after the production.
3. **RealMedia Stream Production** (optional)
  - (a) Select in the leftmost tab the directory of the the AVI file, then in the medium tab the file itself and move it to the Storyboard (the top area). Then select the menu option **File/Produce Movie**.

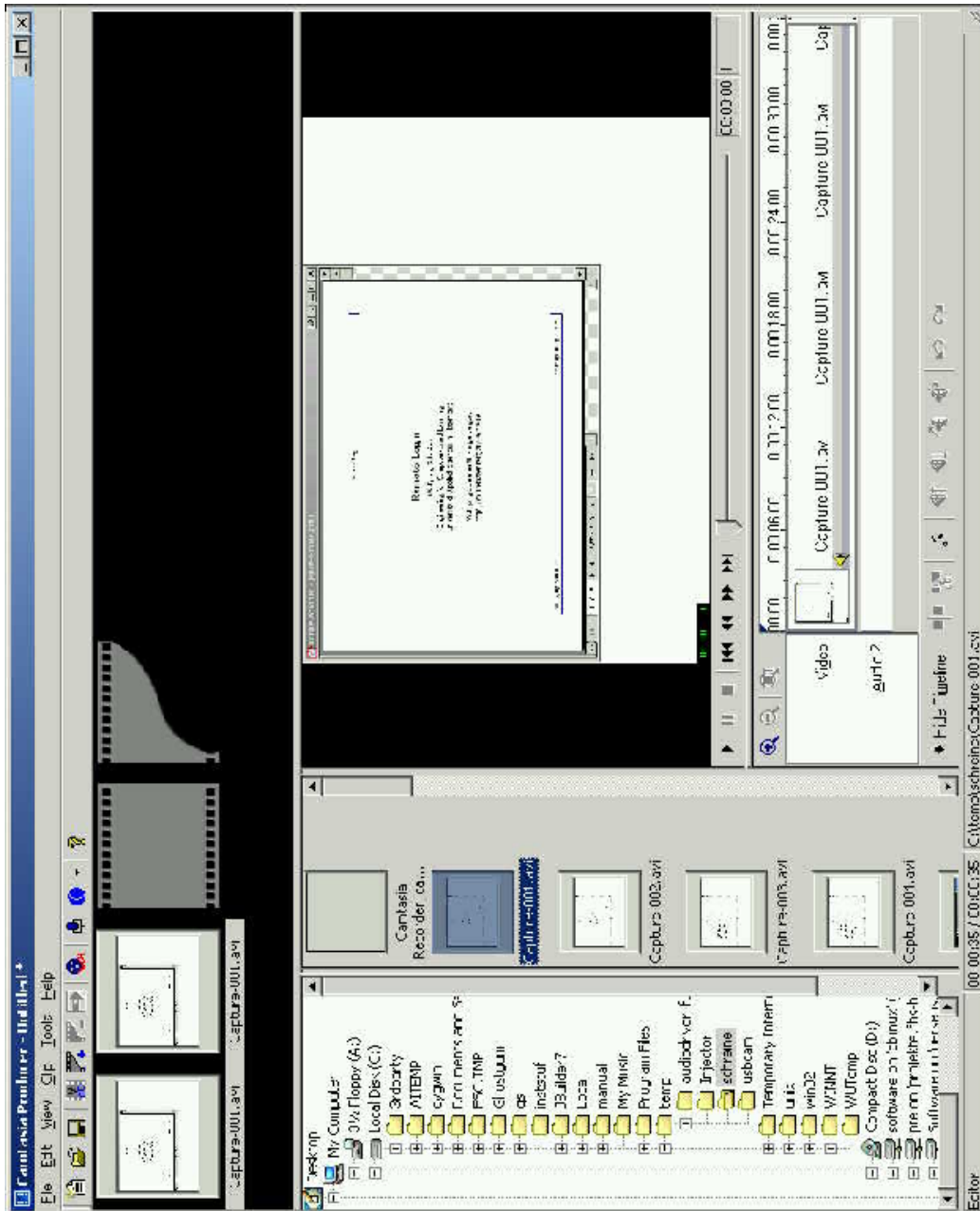


Figure 4: The Camtasia Producer

- (b) Choose as **Movie File Format** the value **RealMedia streaming media file** and press the **Options** key.
  - i. Select as **File Type** the option **Single Rate - Web Server & RealServer G2 compatible** (otherwise the file will not be streamed by a web server but completely downloaded before starting playing).
  - ii. Select as **Target Audience** the value **28K Modem**.
  - iii. Select as **Audio Format** the value **No Audio** and as **Video Quality** the value **Sharpest Image**.
  - iv. Select the options **2-pass Encoding** and **Variable Bit Rate Encoding**.
- (c) Press the button **Produce**; the production process will take about 1.5 times as much as the recording length. The result is a **.rm** file which will be played after the production.

#### 4. **Single Presentation from Multiple Recordings**

It is possible to join multiple recordings to a single presentation, e.g., if a recording had to be interrupted. In this case, it is probably necessary to cut off some trailing part of the recording. The necessary steps are the following:

- (a) Select in the leftmost tab the directory of the the AVI file, then in the medium tab the file itself and move it to the Video Editor (the big central area). You have now the possibility to play the AVI file and to edit it.
- (b) Place the vertical line at the point where you want to cut off the trailing part of the recording. Please note the display of the current time and of the current frame on the right; for a more accurate selection you may stretch the time scale with the magnifier symbol on the left. Press the “Split Video” button (the first button on the bottom row). Both parts of the video are now shown in the Storyboard. Delete the last part.
- (c) Repeat above steps for every recording; as a result the Storyboard holds a sequence of video pieces. Then produce the presentation (Macromedia Flash File and RealMedia stream) as described above.

The results of above steps are a Flash (**.swf**) file and optionally a RealMedia (**.rm**) stream.

## 7 Web Transport

### 7.1 Bandwidth Requirements

The generated Flash file typically contains about 0.5 MB per minute, i.e., a presentation of 30 minutes typically occupies 15 MB (rarely up to twice as much). The Macromedia Flash Player can play a presentation while it is still downloading; for a continuous presentation a network bandwidth of at least 70 kbit/s is required.

The optional RealMedia stream contains less than 0.2 MB per minute, i.e., a 30 minutes presentation occupies about 5 MB. For a continuous presentation of this stream by the RealOne Player a network bandwidth of about 24 kbit/s is required.

The gross bandwidth requirements for both streams together are thus about 100 kbit/s; this is certainly beyond the capabilities of a telephone modem but no problem for broadband technologies based on DSL (a typical ADSL download rate is 512 kbit/s) or cable networks.

Alternatively, the presentations can be transported via a CD-ROM which is able to hold about 16 hours presentations in its 640 MB space.

### 7.2 Web Server

In order to have the various file types served with the correct MIME type, you may have to configure your web server correspondingly.

If you use the Apache server, this can be achieved by placing into (a parent of) the directory being served a file `.htaccess` with the following contents:

```
AddType text/javascript js
AddType application/x-shockwave-flash swf
AddType application/vnd.rn-realmedia rm
AddType application/vnd.rn-realvideo rv
AddType application/vnd.rn-realaudio ra,ram
AddType audio/x-pn-realaudio-plugin rpm
```

### 7.3 JavaScript Controls

To have the player plugins (Macromedia Flash and Real Media) controlled via a Web interface, we use a JavaScript file `flashRealNavigation.js`. For controlling just the Flash Player, we use the file `flashNavigation.js`.

The URLs of these files and their full contents are listed in Appendix D. Place these files in a central location where they can be served by your Web server.

## 7.4 RealMedia File Wrapper

For the generated RealMedia stream (.rm) file, a corresponding .rpm file must be created. This is a text file that contains as its single line the URL of the .rm file, e.g., a file `video.rpm` which contains the line

```
http://cbl.fh-hagenberg.at/~schreine/video.rm
```

The .rpm file is referenced by the HTML file that wishes to have the .rm file played by the RealMedia player plugin.

## 7.5 Presentation File

The presentation itself is an HTML file with the following entities:

- a title displayed in the browser window,
- the Flash presentation,
- (optionally) the RealMedia video,
- a list of buttons and displays for the control of the players.

The HTML file refers to the following files:

- the JavaScript (.js) file for the player control,
- the Flash (.swf) file for the presentation itself,
- the RealMedia (.rpm) file for the video.

These are the values that have to be customized.

See Appendix E for two sample files, one for a Flash presentation together with a RealMedia video, one for a Flash presentation alone.

## 7.6 Index File

To start the presentation `PRESENTATION.html` in a separate browser window of appropriate size and with minimum decoration, a file linking the presentation should contain the following piece of code:

```
<script type="text/javascript">
function openWindow(url)
{
    open(url, "present", "toolbar=no,location=no,directories=no,
        status=no,menubar=no,scrollbars=no,width=1000,height=680");
}
</script>

<a href="#" onClick="openWindow('PRESENTATION.html');">Presentation</a>
```

To start a corresponding presentation without a screen area for a RealMedia stream, the value `width=820,height=680` should be chosen.

## 7.7 Usage Logging

To log the usage of each request to a presentation, we place the Flash movies in a separate directory for which basic authentication is enabled (see the next section). Furthermore, we install a CGI handler `LOG` for this directory which logs the authentication information provided by the user:

```
#!/bin/sh
echo `date` $REMOTE_USER $REMOTE_HOST $PATH_INFO >>LOGFILE
echo "content-type: application/x-shockwave-flash"
echo ""
cat BASEDIR$PATH_INFO
```

`LOGFILE` denotes the file to which the information is logged (must be writable by the server process), `BASEDIR` denotes the path of the server base directory (e.g. `/home/www`).

Using the WN web browser, this CGI handler is installed by placing into the directory to be served a file `index.wn` with contents

```
Attributes=serveall
Default-CGI-Handler=CGIPATH/LOG
```

Here `CGIPATH` denotes the path where the CGI script `LOG` presented above is located.

We then run the command `wndex`.

## Dummy Authentication

To enforce “dummy authentication” (i.e., login name and password have to be entered but are not checked against a password file), we install a dummy authorization software by enhancing the `index.wn` with the following lines and subsequently run the command `wndex`:

```
Authorization-realm=PRESENTATION
Authorization-module=CGIPATH/WNR
Authorization-type=basic
```

Here `PRESENTATION` denotes an arbitrary string that is displayed in the authorization request and that identifies the realm for which the authorization is requested. `CGIPATH` is the path to a directory where the following CGI script is installed:

```
#!/bin/bash
java -cp CLASSPATH WNR LOGFILE2
```

Here `CLASSPATH` denotes the directory where a Java file `WNR.class` is located which implements the command `WNR` (see below). `LOGFILE2` denotes the path of a logfile where the authorization information is recorded.

The command `WNR` enforces authentication and records the information entered; the Java source of this command is shown in Appendix F.

## References

- [1] Regina Ghene and Chriss Jesshope. *Interactive Multimedia for Dummies*. Networked Learning Conference, March 26–28, 2002, Sheffield, UK.  
<http://www2.dcs.hull.ac.uk/people/csscrj/Papers/net-learn-ag.pdf>
- [2] Paul-Thomas Kandzia and Gabriela Maass. *Course Production — Quick and Effective*. 3rd Conference on New Learning Technologies (NLT/NETTIES), September 13–14, 2001, Fribourg, Switzerland.  
<http://www.informatik.uni-freiburg.de/~kandzia/Publications/nlt01.pdf>
- [3] Overview of LecCorder System. Visit: March 19, 2003.  
<http://www.webwisdom.com/products/LecCorder>
- [4] Macromedia Flash MX. Visit: March 19, 2003.  
<http://www.macromedia.com/software/flash>

- [5] Macromedia Flash Player Adoption Statistics. Visit: March 19, 2003.  
[http://www.macromedia.com/software/player\\_census/flashplayer](http://www.macromedia.com/software/player_census/flashplayer)
- [6] R. Müller, T. Ottmann, H. Zhang. *Presentation Recording as a Means to Go Virtual for Campus-based Universities*. Institut für Informatik, Universität Freiburg, Germany, 2002.  
<http://ad.informatik.uni-freiburg.de/~ottmann/publications/irma2002.html>.
- [7] Tegrity Homepage. Visit: March 19, 2003.  
<http://www.tegrity.com>

## A Software Links

The following software has been referenced in this paper:

- Mozilla  
<http://www.mozilla.org>
- Microsoft Internet Explorer  
<http://www.microsoft.com/windows/ie/default.asp>
- Flash Player 6 Plugin  
[http://www.macromedia.com/shockwave/download/triggerpages\\_mmmcom/flash.html](http://www.macromedia.com/shockwave/download/triggerpages_mmmcom/flash.html)
- RealOne Player Plugin  
[http://germany.real.com/freeplayer\\_r1p.html](http://germany.real.com/freeplayer_r1p.html)
- Camtasia Studio  
<http://www.techsmith.com/products/studio>
- Philips VRecord  
<http://www.pc-cameras.philips.com>
- Microsoft Netmeeting  
<http://www.microsoft.com/windows/netmeeting>
- Adobe Acrobat Reader  
<http://www.adobe.com/adobe/acrobat/readstep2.html>
- Actual Title Buttons  
<http://www.actualtools.com/titlebuttons>

## B Camtasia Recorder Configuration

To configure the Camtasia Recorder, perform the following steps:

1. Select the menu option `Effects/Audio/Record Audio`.
2. Select the menu tab `Tools/Options...`
  - (a) Enter the tab `AVI`.
    - i. Enter in `Video Options` 3.0 frames per second. This is sufficient for most purposes but you might want to select a higher value for smoother motions (at the price of a larger file).
    - ii. Press the button `Video Options/Video Setup`.
      - Select as a `Compressor` the `Techsmith Screen Capture Codec`.
      - Press the `Configure` button and select `Better Compression`.
    - iii. Press the button `Audio Options/Audio Setup`
      - Select as `Format` the value `PCM`.
      - Select as `Attributes` value `22.050 kHz; 8 Bit; Mono`. This gives reasonably good quality.
    - iv. Select the `Audio Capture Device` where your microphone is attached (typically your sound card).
    - v. Press the `Volume` button and check that the microphone is enabled and `Volume` is set to high.
  - (b) Enter the tab `File`.
    - i. Select as `Output File Name` the Value `Automatic File Name` and an `Output Folder` and a `Temporary File Folder` on your local disk (not a network disk located on another machine). For every recording, a new file will be created in the denoted output folder.
  - (c) Enter the tab `Hotkey`.
    - i. Select as the `Record/Pause` hotkey `F10` and as the `Stop` hotkey `F11`. This setting does not collide with the key settings of Acrobat Reader to be used for slide presentations.
  - (d) Enter the tab `Program`.

- i. Select the following **Capture Options**: **Pause before starting capture** (which is needed to synchronize both recording streams, see below), **Solid capture rectangle** (to be sure about the boundaries of the captured region), **Play movie after saving file** (to immediately control the quality of the recording), **Play AVI files with Camtasia Player** (in case there is no other player installed), **Round frame size to a 4 pixel boundary** (to minimize the size of the recording), **Display display acceleration during capture** (the performance of the PC during recording is very low otherwise).
3. Select the menu option **Effects/Audio/Record Audio**.
4. Select the menu option **Capture/Input/Fixed Region**.
  - (a) Select the option **Fixed starting point**, choose as **Width** and **Height** the values 800 and 600 (such that the results can be conveniently viewed on a screen with resolution 1024 times 768) and suitable values for **StartX** and **StartY**, e.g., 100 and 100.

## C Wacom Graphire2 Configuration

To configure your Wacom Graphire2 tablet, perform the following steps:

1. Start in folder **Programs/Wacom Tablet** the program **Wacom Tablet**.
2. Select as the tool the **Graphire2 Pen**.
3. Select in tab **Mapping/Speed** as **Positioning Mode** the value **Pen Mode** and as **Display Area** the value **Portion of Display**. Press the button **Click to define screen area** and choose the top left corner and the bottom right corner of the recording area.
4. Select in tab **Tip Feel** the **tip pressure feel** to a **Soft** value (draw in the **Test** area) such that tipping on the tablet is not interpreted as a mouse click (which can become very annoying).

## D Player Control

### D.1 flashRealNavigation.js

The file can be downloaded from <http://cbl.fh-hagenberg.at/~schreine/papers/recordtech2003/flashRealNavigation.js>.

```
var gfactor = 0;

function setClock()
{
    var f = window.document.flash.CurrentFrame()/gfactor;
    window.document.myform.clock.value = String(Math.floor(f)) + "s";
}

function startClock(factor)
{
    gfactor = factor;
    setInterval("setClock();", 1000);
}

var tid = 0;

function startMovie()
{
    if (window.document.realstream.GetPlayState() == 0)
    {
        clearInterval(tid);
        tid = 0;
    }
    else if (window.document.realstream.GetPlayState() == 3)
    {
        clearInterval(tid);
        tid = 0;
        window.document.flash.Play();
    }
}

function fstart(factor)
{
    if (window.document.realstream.GetPlayState() == 0)
        frewind();
    window.document.realstream.DoPlay();
    tid = setInterval("startMovie();", 200);
}

function fstop()
{

```

```

    window.document.flash.StopPlay();
    window.document.realstream.DoPause();
}

function frewind()
{
    window.document.flash.Rewind();
    window.document.realstream.DoPause();
    window.document.realstream.SetPosition(0);
    setClock();
}

function setFrame(frame, factor)
{
    if (frame < 0) frame = 0;
    var IE = navigator.appName.indexOf("Microsoft") != -1;
    if (!IE)
    {
        var n = window.document.flash.TotalFrames();
        if (frame > n-2) frame = n-2;
    }
    window.document.flash.GotoFrame(frame);

    if (window.document.realstream.GetPlayState() == 0)
    {
        window.document.realstream.SetPosition(0);
        window.document.realstream.DoPlay();
    }

    window.document.realstream.DoPause();
    var ms = (1000*frame)/factor;
    var len = window.document.realstream.GetLength();
    if (ms >= len) ms = len-1;
    if (ms < 0) ms = 0;
    window.document.realstream.SetPosition(ms);
    setClock();
}

function fforward(factor)
{
    var i = parseInt(window.document.myform.mytext.value);
    var f = window.document.flash.CurrentFrame();
    setFrame(f+factor*i, factor);
}

function fback(factor)
{

```

```
var i = parseInt(window.document.myform.mytext.value);
var f = window.document.flash.CurrentFrame();
setFrame(f-factor*i, factor);
}
```

## D.2 FlashNavigation.js

The file can be downloaded from <http://cbl.fh-hagenberg.at/~schreine/papers/recordtech2003/flashNavigation.js>.

```
var gfactor = 0;

function setClock()
{
  var f = window.document.flash.CurrentFrame()/gfactor;
  window.document.myform.clock.value = String(Math.floor(f)) + "s";
}

function startClock(factor)
{
  gfactor = factor;
  setInterval("setClock();", 1000);
}

function fstart()
{
  var movie = window.document.flash;
  movie.Play();
}

function fstop()
{
  var movie = window.document.flash;
  movie.StopPlay();
}

function frewind()
{
  var movie = window.document.flash;
  movie.Rewind();
  setClock();
}

function fforward(factor)
{
  var movie = window.document.flash;
  var i = parseInt(window.document.myform.mytext.value);
```

```

    var f = movie.CurrentFrame();
    f = f+factor*i;
    movie.GotoFrame(f);
    setClock();
}

function fback(factor)
{
    var movie = window.document.flash;
    var i = parseInt(window.document.myform.mytext.value);
    var f = movie.CurrentFrame();
    f = f-factor*i; if (f < 0) f=0;
    movie.GotoFrame(f);
    setClock();
}

```

## E Presentation File

### E.1 Flash Presentation with RealMedia Stream

The file can be downloaded from <http://cbl.fh-hagenberg.at/~schreine/papers/recordtech2003/exampleFlashReal.js>.

```

<html>
<head>
<title>PRESENTATION TITLE</title>
</head>
<body bgcolor="#DDDDDD">

<table>
<tr valign=top>

<td>
<embed name="realstream" id="realstream"
    src="PRESENTATION.rpm"
    width=160
    height=120
    backgroundcolor=white
    console=_unique
    controls=ImageWindow
>

<td>
<object id="flash" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase=

```

```

    "http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=4,0,0,0"
    width="801" height="601">
<param name="movie" value="PRESENTATION.swf">
<param name="quality" value="high">
<param name="bgcolor" value="#000000">
<embed name="flash" src="PRESENTATION.swf" quality=high bgcolor=#FFFFFF
    width=801 height=601 type="application/x-shockwave-flash"
    pluginspace="http://www.macromedia.com/go/getflashplayer">
</embed>
</object>

</table>

<script src="FLASHREALNAVIGATION.js" type="text/javascript">
</script>

<p>
<form name="myform">
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="button" name="startb" onclick="fstart(3)" value="Play">
<input type="button" name="stopb" onclick="fstop()" value="Pause">
<input type="button" name="rewindb" onclick="frewind()" value="Rewind">
</span>
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="button" name="forwb" onclick="fforward(3)" value="Forward">
<input type="button" name="backb" onclick="fback(3)" value="Back">
<input type="text" name="mytext" value="60" size=1
    style="text-align: right;"> s
</span>
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="text" name="clock" value="0s" size=3 readonly=1
style="border-style: none; background: #EEE; text-align: right;">
</span>
</form>
<script>
startClock(3);
</script>

</body>
</html>

```

The values PRESENTATION TITLE (title in the browser window), PRESENTATION.rpm (location of the .rpm file), PRESENTATION.swf (location of the .swf file, two occurrences!), and FLASHREALNAVIGATION.js (location of the .js file) have to be appropriately customized. The argument “3” in the calls of various JavaScript functions denotes the number of frames per second in the Flash file.

## E.2 Flash Presentation

The file can be downloaded from <http://cbl.fh-hagenberg.at/~schreine/papers/recordtech2003/exampleFlash.js>.

```
<html>
<head>
<title>PRESENTATION TITLE</title>
</head>
<body bgcolor="#DDDDDD">
<object id="flash" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase=
  "http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=4,0,0,0"
  width="801" height="601">
<param name="movie" value="PRESENTATION.swf">
<param name="quality" value="high">
<param name="bgcolor" value="#FFFFFF">
<embed name="flash" src="PRESENTATION.swf" quality=high bgcolor=#FFFFFF
  width=801 height=601 type="application/x-shockwave-flash"
  pluginspace="http://www.macromedia.com/go/getflashplayer">
</embed>
</object>

<script src="FLASHNAVIGATION.js" type="text/javascript">
</script>

<p>
<form name="myform">
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="button" name="startb" onclick="fstart()" value="Play">
<input type="button" name="stopb" onclick="fstop()" value="Pause">
<input type="button" name="rewindb" onclick="frewind()" value="Rewind">
</span>
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="button" name="forwb" onclick="fforward(3)" value="Forward">
<input type="button" name="backb" onclick="fback(3)" value="Back">
<input type="text" name="mytext" value="60" size=1
  style="text-align: right;"> s
</span>
<span style="border: outset; margin: 3; padding: 6; background: #EEE">
<input type="text" name="clock" value="0s" size=3 readonly=1
style="border-style: none; background: #EEE; text-align: right;">
</span>
</form>
<script>
startClock(3);
</script>
```

```
</body>
</html>
```

The values `PRESENTATION TITLE` (title in the browser window), `PRESENTATION.swf` (location of the .swf file, two occurrences!), and `FLASHNAVIGATION.js` (location of the .js file) have to be appropriately customized. The argument “3” in the calls of various JavaScript functions denotes the number of frames per second in the Flash file.

## F Dummy Authentication

The file can be downloaded from <http://cbl.fh-hagenberg.at/~schreine/papers/recordtech2003/WNR.java>.

```
import java.io.*;
import java.util.*;

public class WNR
{
    // read http authentication header and log information in a file
    // returns 0 on granting access, 1 on denying access
    public static void main(String[] args)
    {
        String string = null;

        try
        {
            // read authentication string
            BufferedReader reader =
                new BufferedReader(new InputStreamReader(System.in));
            string = reader.readLine();
            reader.close();
        }
        catch(Exception e)
        {
            System.out.println("Error when reading line");
            System.exit(0);
        }

        // remove header
        if (!string.startsWith("Basic "))
        {
            System.out.println("wrong string");
            System.out.println(string);
        }
    }
}
```

```
        System.exit(1);
    }
    string = string.substring(6);

    // open file for appending
    String fname = args[0];
    Writer writer = null;
    try
    {
        writer = new FileWriter(fname, true);
        writer.write(new Date().toString());
        writer.write("\t");
        writer.write(string);
        writer.write("\n");
        writer.close();
    }
    catch(Exception e)
    {
        System.out.println("Error when writing file");
        System.exit(0);
    }

    System.exit(0);
}
}
```