Synchronization of Video Streams in the Implementation of Web-Based E-Learning Courses

KARL KURBEL, ALEXEI PAKHOMOV
Chair of Business Informatics
Europe University Viadrina
Grosse Scharrnstrasse 59, D-15230 Frankfurt (Oder)
GERMANY

Abstract: E-learning courses based on video-streaming technology are a promising form of transferring lectures to the Internet. Though not applicable to all types of courses, lecturing is a well-tried educational mode suitable in many cases and proven for transporting content. In this paper an approach is discussed in which course material is divided into separate parts – one silent video file containing visual demonstrations, and another video with sound, with the instructor's explanations. We discuss approaches to achieve synchronization among two separate videos streams. Based on Microsoft Windows Media technology this problem can be solved but the solution is not straightforward. The solution is based on integration of two Windows Media Player ActiveX controls in one HTML page. Problems encountered and the final solutions to those problems are outlined.

Keywords: Streaming video, video-based e-learning courses, Windows Media technology, synchronizing playback, Virtual Global University

1 Video as a Basis for E-learning

Web-based teaching and learning are becoming more and more popular as the numbers of virtual courses and virtual universities (like JIU [1], VGU [2], and VIROR [3]) are growing. While in the beginning static HTML pages, text, and graphics files were mainly used, animated and multimedia contents have significantly enhanced e-learning since then. Video clips are also found in web-based courses, but videos as the primary instructional medium are not very common yet. One reason for that is certainly that production and delivery of video-based courses are rather time consuming, requiring profound know-how about the technology.

On the other hand, video-based courses have one significant advantage. Many lecturers have years of experience giving "lectures", that is standing in a lecture hall or a classroom and explaining complicated matters to students, possibly using visual aids like overhead transparencies or PowerPoint slides. In computer-oriented subjects, online presentations on a video beamer may be another form of visualization. Using videos as the main instructional medium makes it possible to transfer this well-tried, though traditional method of teaching to the web age.

Common platforms to provide and distribute streaming video content over the Internet are Microsoft Windows Media [4], Real Media [5], and Apple QuickTime [6].

The authors have chosen Microsoft Windows Media technology for several reasons. First, the Windows Media Video V7 Screen codec (codec = coder/decoder for video and audio data) proved to be very good for online recordings from a computer screen. In our courses we need this feature because often things are explained by the lecturer online and real-time. Second, the Microsoft Media Server is included in the licence for Microsoft Internet Information Server (IIS) and that licence does not limit the number of client connections. The Real Media Server, for example, is free but only for 25 client connections at a time.

In the subsequent section, the front-end of a video-based e-learning course as it appeares to the user is discussed. Section 3 addresses the problems that have to be attacked to produce and deliver such a course over the Internet. In section 4, features and shortcomings of the Windows Media Player as observed in our work are described and the solution to the problems encountered is presented. Some conclusions from our experience with video-based courses are drawn in section 5.

2 E-Learning User Interface with Two Video Streams

Following the paradigm of lecturing and visualizing, the user interface of the Virtual Global University's video-based courses as it is presented to the student is shown in figure 1. The crucial part of the web page is the Media Player in the upper right corner. In this corner a video of the lecture is running in a streaming compatible format.

The area left of the player is used to show visual materials accompanying the lecturer's explanations. In the simplest case these are text slides, tables, diagrams, charts, etc., similar to information that would be written on a blackboard or an overhead projector in a conventional lecture. In addition there are controls for user interaction with the course page, for example starting the lecture, advancing or going back to a certain topic, etc.

Fig. 1: User interface of a video-based e-learning course by VGU [2]

Basically there are two different approaches to provide visual material plus video lecturing for such a course. The first way which is fairly easy to achieve is to split up the lecture in separate topics and store the respective video clips in separate files. If the topics are associated with slides, then each topic, including the video clip, will be started separately.

Although this solution is easy from a technical point of view it is not always feasible to split up a lecture in such a way. Moreover, only static images can be used as accompanying visual information. Displaying online recordings of program runs, for example, or videos shown by the lecturer is not feasible in this way.

One possible solution for this problem could be to use a pop-up window in which the second video runs. However, this is inconvenient, requiring time for connecting to the media server and for preloading/buffering some video data.

In some situations it is better to use two synchronized video clips. Sometimes this is even the only possible solution. One example is the production of multilingual content. Here it is advisable to separate a video file with visual material from sound files in which different languages are represented. This solution saves space on the hard disk, proportional to the number of languages and to the size of the video file.

Another case is when recordings from two or more video sources have to be displayed at the same time. For example, it is not possible to use screen capturing on two ore more computers and record the pictures into one file.

One more problem is that screen recordings cannot be merged with recordings from a video camera without serious loss of quality, or only at the cost of very large file sizes.

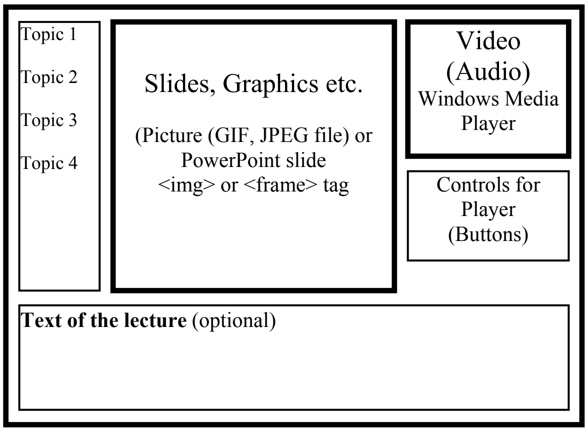

Fig. 2: Functional diagram of a course page

In the general outline of a course page as shown in figure 2 the area for visual material can be used for displaying a video stream as well. How this is achieved will be discussed in section 4. For the student there is no difference in the graphical user interface no matter whether static files or videos are displayed. Technology dependent peculiarities are hidden below the user interface.

3 Integrating Separate Video Streams: Fundamental Problems

Major problems in creating course pages like the ones in figures 1 and 2 result from limitations of the Windows Media Player object model. The available methods and properties do not provide mechanisms to control all stages of client-server interaction and to redistribute connection bandwidth among two players adequately. In other words, even if more connection bandwidth is available than required by the sum of bit rates of the two video streams, one cannot expect that the bandwidth will be divided in proportion to the bit rates of the streams, nor even equally.

This shortcoming implies that two video files will rarely start playing simultaneously even if they have exactly the same parameters (e.g. resolution, codec, file size). The synchronization task is thus left to the course creator.

To guarantee proper appearance and behavior of video-based courses we developed and implemented an algorithm for synchronizing and monitoring two video streams. It has the following components as shown in figure 3:

- Synchronizing playback start,
- Monitoring synchronized playback of two videos,
- Corrections if synchronization is off,
- Setting two players to specific positions on demand.

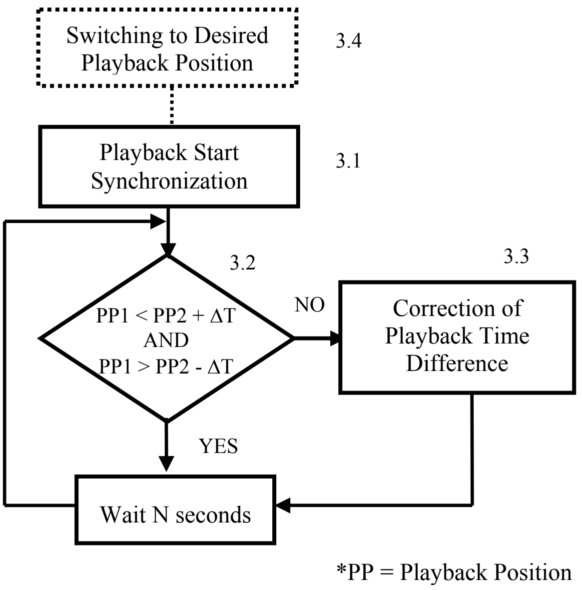

Fig. 3: Synchronization algorithm

3.1 Synchronizing the Start of Two Players

Since two players hardly ever start at the same time a method of ensuring synchronized start is needed. The source of the problem are actually different data compression algorithms in the codecs used.

One video file in a course page as outlined above originates from screen capturing with Matchware Screen Corder 2.0 [8] and is translated into streamable format with the Windows Media Video V7 Screen codec. The conventional camera recording is compressed with the Windows Media Video V8 codec.

Because of the particularities of these codecs a procedure is needed which checks the states of the players. When the first one is ready to play a video, the procedure checks the state of the second player and waits until the second one is also ready to start.

3.2 Monitoring Synchronized Playing

Once playback has started it is necessary to check at regular, possibly short intervals if the two videos are still in the correct positions. The playing times can be measured in seconds or frames from the start of the files. If a difference is detected and if that difference exceeds a certain limit a procedure is started that eliminates the time lag of one player.

3.3 Correcting Differences in Synchronization

This procedure pauses the player that is running ahead and checks playback of the late player. This is done until the slower player has approximately caught up with the other one. Then the first player is released to continue playing.

One serious problem, however, needs to be observed. Due to various reasons the players can change their states (like playing, pausing, buffering, etc.) between individual calls of the procedure and even during the execution of the procedure. Reasons for this can be, for example, actions by the user, reaching a marker in the video file, or a state change in the connection with the Media Server.

Therefore additional monitoring of the states of the two players not only at the beginning but also during playback is very important. Otherwise the procedure can crash or the user's web browser can get frozen.

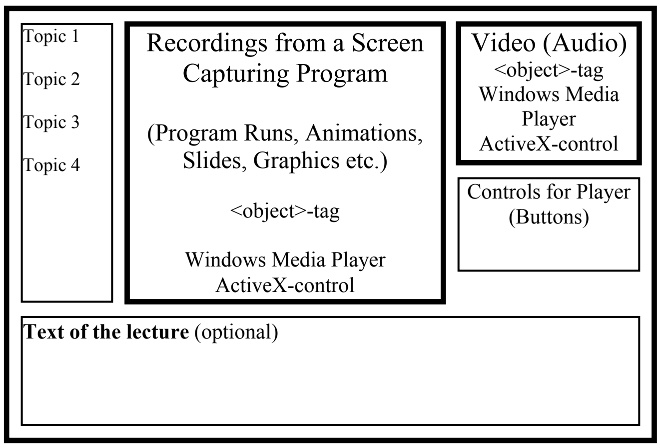

Fig. 4: Course page after embedding a second ActiveX control

3.4 Setting two Players to Same Position

As indicated in figure 2 the user may jump directly to a specific topic of the lecture forwards or backwards by clicking on a control on the left-hand side. In such a case both video files have to be advanced or set back to the desired positions in a synchronized way.

Once this is achieved both players have to continue or start playing at the same time.

This is basically the same problem as ensuring a synchronized start of two players in the first procedure (see section 3.1), so that procedure can be extended and reused.

4 Integrating Video Streams with Windows Media Technology

Providing another video on the course page, in addition to the video showing the lecturer, made it necessary to find a way to include two instances of the Windows Media Player (WMP) into one page. As stand-alone applications, two or more players cannot be started. Microsoft's technical documentation, on the other hand, does not specify whether it is allowed to use two or more WMP ActiveX controls in the same HTML page or not.

Therefore tests were performed to find out whether this is possible or not. Two WMP ActiveX controls were embedded in one page. In fact, both controls established a connection to the Media Server and started playback whenever the page was loaded. At the same time we found that a synchronization procedure is absolutely necessary because the playbacks never started simultaneously.

From the currently available versions of Windows Media Player, 6.4 and 7.1 [7], the latter one was chosen for several reasons. Compatibility of those players with various operating systems and internet browsers is shown in table 1.

Operating System					
Version	Win 95	Win 98	WinNT	WinMe	2000/XP
WMP 6.4	X	X	X	X	X
WMP 7.1		X		X	X

Internet Browser			
	Microsoft	Netscape	
Version	IE 5.x,6x	Navigator 4.x	Navigator 6.x
WMP 6.4	X	With Plug-In	X
WMP 7.1	X		

Tab. 1: Compatibility of Windows Media Player versions 6.4 and 7.1

The following points were taken into consideration for the decision which WMP version to use:

- The object model of version 7.1 has new properties and methods that can be used in scripts to manage client-server connections effectively.
- Although WMP 7.1 is not available for Windows NT, this is normally no problem because Windows NT is rarely used on client computers.
- WMP 6.4 is available for all Windows operating systems but not on other platforms like the Mac platform.

From these considerations, the version 7.1 Windows Media Player seemed to be better suited to obtain cross-platform compatibility. For full compatibility with all Windows version WMP 6.4 has to be used. If the user has not defined Microsoft Internet Explorer (IE) as the standard browser it is always possible to start IE explicitly for playing streaming videos.

In our initial work with WMP 7.1 two ActiveX controls where embedded into the HTML page with the help of the <object> tag. Synchronization procedures for playback as outlined above were developed in VBScript (Visual Basic Script Edition).

Testing the initial pages on six computers with different processors and connection speeds showed that program crashes occurred at some times. The number of failures was inversely related with actual system performance. The number depended not only on the configuration of the computer but also on the current system usage by other applications. There was no indication that the number of failures depended on the connection bandwidth.

Since the reasons for those system failures were neither obvious nor found in the technical documentation, a testing procedure had to be developed to allow for step-by-step execution of the synchronization procedures. Normal script debugging techniques – turning off all "error tolerance" options in the Internet browser and implementing break points with message box alerts – was not appropriate because message boxes produce modal dialog windows. So at each alert the script procedures would have paused until the dialog window has been closed.

Therefore an "alert system" based on frames in HTML pages was developed. In a frame the content can change dynamically without affecting other frames. The test page with an additional frame is shown in figure 5. The course page to be tested is contained in the first frame. The second frame is used to display information about the currently executed procedure or command.

In a sequence of tests it was found that all page crashes and browser hang-ups are occurring during or immediately after the first access to the properties or methods of the Media Player ActiveX control. However, when that control was accessed again after some time crashes rarely occurred.

This behavior of the WMP 7.1 control gave rise to the assumption that for some reason the indicated state of a control (i.e. the value of the "PlayState" property) changes faster then the actual state of that control. In other words, the value of the properties is changed first and only afterwards all operations belonging to that state change are executed. This observation explains to some extent why the scripting procedures behave differently when executed on different computers.

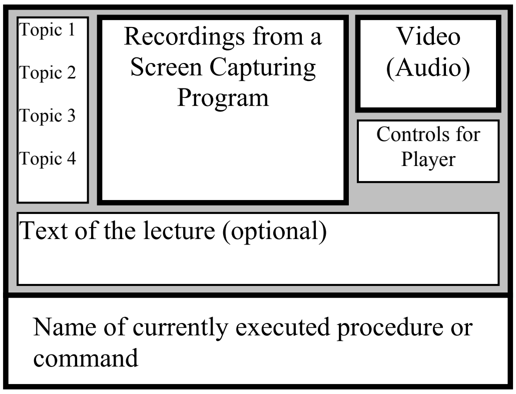

Fig. 5: Test-page layout for debugging of synchronization procedures

A similar phenomenon can be observed on a conventional video tape recorder: When the "Play" button on the remote control is pressed the "Play" sign appears immediately on the recorder's display. However, before a picture from the tape appears on the screen the tape has to be stretched around the video head, and appropriate speeds of both revolving of the head and transporting the tape have to be established. The process may take up to several seconds. During that time the recorder displays the "Play" sign but no picture is visible on the screen.

This problem has been known in the field of professional video editing for a long time. Today's video equipment has special functions that allow winding the tape back for several seconds to compensate for the difference between the indicated and the actual playback start.

Solving the problem of delayed action in the underlying problem in an analogous way was not possible. Although placing pauses after each access to the WMP properties would have let the player equalize its indicated and its actual states, determining the length of the pauses posed a major problem. Whereas on a 1440 MHz system all procedures worked properly and without delay when the properties of the player control were accessed, on a 700 MHz, 128 MB RAM system the delay needed to ensure stable behavior of the procedures amounted to at least four seconds.

During this relatively long time changes of the states of both player controls can occur – due to user actions, to events caused by video playback, or to events in the network connection. Those events can happen in any arbitrary manner. While user actions and playback events could theoretically be "blocked" or at least their handling postponed, this is not possible for events related with the media server or more generally with the network connection. Those events affect not only the HTML page and its event watching procedures but also directly the playback of a video stream. If the actual connection speed becomes too slow to deliver both video streams simultaneously, at least one of them will pause for additional buffering in the background regardless of what the synchronization procedures are doing.

Even if it were possible to protect the state of a WMP control from changes caused by events and effects except those coming from the synchronization procedure itself, the time needed for correction would have summed up to more than 12 seconds. The reason for this is that the procedure for correcting the time difference requires at least three accesses to the WMP control. Waiting 12 seconds or more for the demonstration or explanation of visual material on the screen was found unacceptable for the e-learning courses under discussion.

The findings from our testing and debugging efforts were that the Windows Media Player 7.1 version requires powerful computer systems to avoid the synchronization problems described above. On the majority of today's PCs the system performance demanded to play synchronized videos in a reliable and stable way is not available. Therefore the decision was made in our project to go back to the Windows Media Player version 6.4in order to obtain reasonable performance of the video material on average personal computers.

Adapting the synchronization procedures for WMP 6.4 required some deep changes because the object models underlying the versions 7.1 and 6.4 exhibit significant differences [7]. For example, the synchronization procedures responsible for playback start use the "BufferingProgress" property of the 7.1 version. This property helps to detect if a player is ready for playback or not. Once the value reaches 100 (%) then it is possible to start the playback. In the 6.4 version, however, this property always gets the value 100 when the specified time for buffering is over, no matter whether the buffering has really been accomplished or not. Therefore the procedure that checks the buffering progress had to be substituted by a procedure that watches the "BufferingStart" event. This event occurs at the start and the end of buffering.

Tests with the WMP 6.4 version were performed on the same computers as the previous tests with WMP 7.1. Results were excellent with respect to system stability, independently of actual system workload and bandwidth of network connection.

5 Conclusions

As a result of our work it was found that video streams can be synchronized using components of the Windows Media platform although this is not specified in the technical documentation.

One solution to the synchronization problem is to use an HTML page with two ActiveX controls representing Windows Media Players. The 7.1 version of the Windows Media Player with many new features is only appropriate if it can be run on a sufficiently powerful computer to avoid the problems and program crashes encountered in our tests.

WMP 6.4 is compatible with all Windows operating systems including Windows NT, and with the Netscape Navigator web browser. This version plays two synchronized video streams correctly, practically on every PC that fulfills the minimum system requirements imposed by modern video codecs such as Windows Media Video V8 and Microsoft MPEG 4 V 3.

The synchronization procedures written in VBScript work correctly and robustly on any PC with the capacity just mentioned. They provide satisfactory playback synchronization even if the available network bandwidth is too low to deliver both video streams simultaneously in real-time. In such a case some pauses will inevitably occur, due to additional buffering, but the videos will still run correctly. If it is known that the network bandwidth is smaller than the sum of the bit rates of the two video streams, then it might be considered to preload one of the files, store it, and synchronize playback with the video stream delivered real-time. The synchronization procedure does not require any changes in this case.

Taking into account that ADSL connections with stable bandwidth of 400 - 700 Kbit/s are increasingly becoming available, this bandwidth is sufficient for uninterrupted delivery of a lecture video with 320 x 240 pixel resolution and 25 fps, compressed with Windows Media Video V8 codec, together with another video containing visual demonstration material with 800 x 600 pixel resolution and 10 fps, compressed with Windows Media Video 7 Screen codec. From our experience it can be concluded that the technical conditions for producing and delivering e-learning courses based on synchronized video streams already exist. In the project underlying our work – developing multimedia courses of the Virtual Global University (VGU) [2] – this approach has been successfully applied.

References:

[1] Jones International University (JIU); http://jiu-web-a.jonesinternational.edu/

[2] Virtual Global University (VGU); http://www.vg-u.org

[3] Virtual University in the Upper Rhine Valley (VIROR); http://www.viror.de

[4] Microsoft Windows Media; http://www.microsoft.com/windows/windowsmedia/default.asp

[5] Real Media; http://www.realnetworks.com/

[6] Apple Quick Time; http://www.apple.com/quicktime/

[7] Windows Media Player; http://www.microsoft.com/windows/windowsmedia/players.asp

[8] Matchware Screen Corder; http://www.matchware.net/

[9] Feamster, Nicolas G., *Adaptive Delivery of Real-Time streaming Video, M. Eng. Thesis,* Massachusetts Institute of Technology, Cambridge 2001

[10] Wee, Susie J.; Apostolopoulos, John G., *Secure ScalableVideo Streaming For Wireless Networks,* IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, Utah 2001