Interactive Content Overviews for Lecture Recordings

Robert Mertens
Virtual Teaching Support Center
University of Osnabrück
Germany
Robert.Mertens@uos.de

Markus Ketterl
Virtual Teaching Support Center
University of Osnabrück
Germany
Markus.Ketterl@uos.de

Oliver Vornberger
Dept. of Computer Science
University of Osnabrück
Germany
oliver@uos.de

Abstract

Lecture recordings can be a powerful addition to traditional lectures and they can even serve as a main content source in a number of didactic scenarios. If users can quickly locate relevant passages in a recording, the recording combines the ease of search that comes with electronic text based media with the authenticity and wealth of information that is delivered in a live lecture. Locating relevant passages in a time based media such as a recorded lecture is, however, not as easy as searching an electronic text document. This paper advocates for interactive content overviews as an efficient navigation aid and presents an implementation that is based on SVG. It also discusses the feasibility of likewise approaches in other media formats.

1. Introduction

Recording live lectures has proven to be a fast and efficient way of creating content for e-learning [11]. This content can be used in a number of ways ranging from simply augmenting a lecture to profoundly changing the didactic setting (see [14] for an overview).

A number of application scenarios aim for replacing the lecture in its traditional form by more interactive learning sessions [4, 5, 8]. In these scenarios, the recorded talk takes over the role of the classic lecture in that it becomes the main means of knowledge transfer.

The way in which knowledge transfer happens in a traditional lecture can however be considered highly inefficient. Traditional lectures deliver content in a one-size-fits-all manner that wastes time of more advanced learners and that often is too fast for those who do not possess the right amount of prior knowledge.

To fully exploit the possibilities given by electronic lectures, advanced navigation features are crucial [12]. Students must be able to skip unnecessary passages of a recording and to repeat problematic sections. This statement holds true not only for novel application scenarios but also when lecture recordings are used as an addition to a conventional lecture.

The aim of this paper is to show how navigation in web lectures can be improved using interactive content overviews for both time based and structure based navigation and how these overviews can be implemented in different media formats such as Flash and Scalable Vector Graphics (SVG).

The paper is organized as follows: Section 2 presents a brief analysis on how navigation in web lectures affects learning efficiency. Section 3 gives an overview of related work in the direction of interactive content overviews in lecture recordings. Section 4 describes an SVG based approach that links an interactive time based overview with an interactive structure based overview. Section 5 discusses implementation details and section 6 shows the potential of an alternative Flash based solution.

2. The Need for Navigation

It is widely believed that learning is most efficient when both the speed in which the learning content is presented and the level at which it is presented is tailored to the learner [3]. This is, however, rarely the case in any traditional lecture.

In contrast to classic lectures, web lectures bring the advantage that the material presented in the lecture can be chunked into small pieces that can be replayed as often as required [7].

In [8] this fact and the possibility to skip known knowledge have been shown to be the key advantages of eLearning. The comparison in figure 1 schematically depicts the effects these advantages can have on how learning time is used. Skipping parts that contain information that is already known to the user and replaying specific parts of a recording does, however, require advanced navigation facilities.

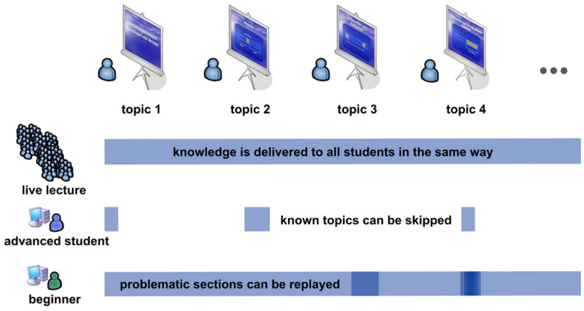

figure 1: schematic comparison of knowledge reception in traditional and web lectures

Experiences described in [16] and [5] as well as a student survey published in [4] back this hypothesis for scenarios in which recorded lectures are used to replace the live lecture.

Results from a study of viewing behaviour in a context in which recordings were used as a supplement to a live lecture indicate that navigation seems to play an important role in this kind of scenarios, too [17].

3. Related Work

Since lecture recording has emerged as a technology for content production in the mid-nineties, a number of navigation facilities for lecture recordings have been developed. These range from simple tables of content to sophisticated navigation aids that incorporate previewing functionalities or similarly advanced features (see [13] for an overview).

A relatively small number of approaches feature interactive content overviews that facilitate navigation by enhancing either time or structure based navigation.

eClass [6] introduces an approach called linking by interacting in which handwritten annotations that are added to a slide during the lecture are added as mouse sensitive elements in the slide overview. Clicking on these elements results in the viewer starting playback at the time index associated with the drawing of the element.

With hierarchical brushing, the timeline of the interface gives a preview of the slide shown at any segment brushed over by the mouse pointer [15]. To provide for a finer granularity, segments can contain a number of sub-segments.

In AOF, the timeline can be used to visually scan the slides of the recording by moving the slider knob using a technique called random visible scrolling [11]. More recent versions employ different techniques that allow for varying scrolling speed in order to achieve a better way to navigate at different levels of granularity [9, 10].

4. Improving Navigation with Interactive Content Overviews

With recent developments in AOF and with hierarchical brushing, time based navigation in lecture recordings has risen to a very high level. It can however be further improved by adding structural context information like the beginning and end of the animation step belonging to the current slider position to the timeline [13]. In the virtPresenter viewer interface, this information is presented dynamically. When the mouse pointer is dragged over the timeline, the boundaries of the surrounding animation step and slide are marked by vertical lines as shown in figure 2. This way, the next animation step or slide can easily be located.

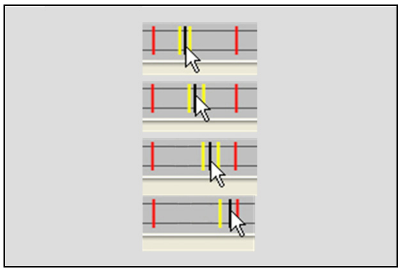

figure 2: dynamic visualization of slide and animation boundaries

In combination with a brushing technique that is similar to visible scrolling and visualizes the corresponding slide in the respective animation state in the overview, this technique has shown to facilitate navigation considerably. The combination of these two approaches is shown in figure 3.

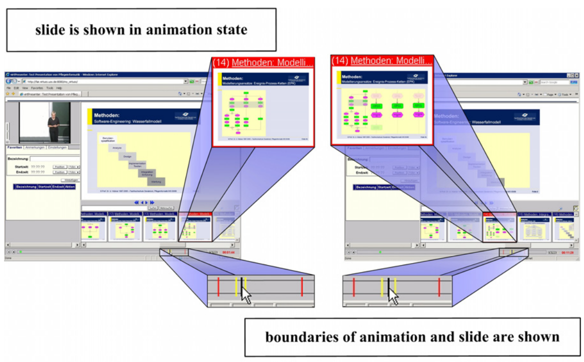

figure 3: navigation in virtPresenter

Another interactive property of the timeline is a time based implementation of the footprint concept known from hypermedia navigation. Whenever replay is started, an animation is started that visually marks the viewed passages on the timeline.

In the thumbnail slide overview, slide elements that were animated during the presentation are mouse sensitive and clicking on these elements starts playback at the time index at which the respective element has first entered the presentation. An important difference between this approach and the linking by interacting approach implemented in eClass is that the slide elements are represented symbolically and can thus be connected to full text search in case they contain text. They can also easily be animated for visual highlighting and elements can be connected to code that is triggered when the mouse pointer is hovered over them. In virtPresenter this property is used to highlight the time index corresponding to a slide element in the timeline as shown in figure 4.

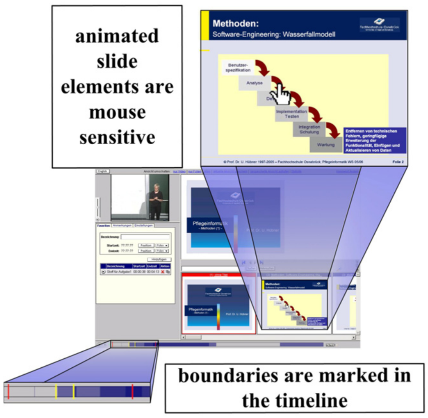

figure 4: interactive slide elements

Another advantage of linking slide elements over linking handwritten annotations is that the lecturer's presentation style can remain unchanged. In an evaluation of eClass, students have preferred presentations that consisted of prepared PowerPoint slides over those that consisted of handwritten notes only [2]. This might serve as an indicator that students prefer slide lectures that are usually easier to read and that are prepared more thoroughly than an impromptu presentation.

Adding script code to slide elements is, however, a non-trivial operation. In the original linking by interacting approach, handwritten annotations could be realized by simple overlays [6]. In virtPresenter, PowerPoint slides are converted to SVG using SVGmaker (www.svgmaker.com). During this process, information about which element of a slide was animated is lost. The following section describes an approach that identifies the animated elements from a PowerPoint slide in an SVG document generated from this slide.

5. Generating the Slide Overview

To identify those elements in an SVG slide to which script code should be added, each animated element of the respective PowerPoint slide is printed in a single SVG document (reference document). In a second step, the content of each SVG document is compared to the content of the SVG document generated from the original PowerPoint slide. Whenever an element is found in the main slide that is equal in appearance to an element found in one of the reference documents the element is enclosed in a scripted <g> node that contains the script code for this element as shown in figure 5.

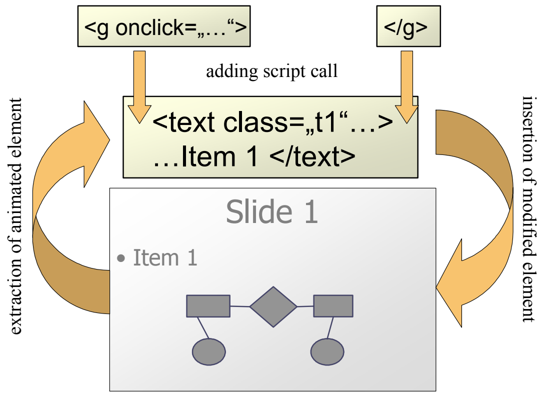

figure 5: script code is added to existing SVG

Since SVGmaker does not necessarily convert equal elements in different PowerPoint slide into SVG elements that are syntactically equal, the comparison algorithm has to check for elements that are rendered in the same way, not for elements that are described the same way in the SVG documents.

SVGmaker generates SVG in a way that keeps files small. In order to do this, the software uses a number of possibilities given by the SVG specification. For instance unconnected paths of the same color are stored in one single path object. These path objects can, however, also contain a number of polygons that are described by paths. As paths describing polygons use relative coordinates and regular paths use absolute coordinates, converting a path object containing a number of paths into one path object for each path requires some parsing and calculation.

Pictures are embedded by the SVGmaker using <use> tags that reference parts of the pictures using xlinks that point to <image> tags containing the actual picture information. This saves space in case an image can be found more than one time in the document. In the comparison algorithm, the content of each <image> tag and its ID are stored in hashtables. When two <use> tags are compared, their coordinates and the respective entries in the hashtables are checked.

Text fonts are specified by references to a CSS which is embedded in a CDATA section in the document. Since the IDs for the same fonts can vary from document to document, the CDATA sections are parsed and IDs and fonts are stored in hashtables again.

After these steps are performed, both documents can be compared element by element. As text elements are not brought down to atomic units (in this case letters), comparing text elements is done using string operations to detect if text elements have regions in common.

If other media formats are to be used for representing the slides, different algorithms must be developed. Alternatively, markers that allow to identify animated elements would have to be integrated into the conversion process.

5. Alternative Media Formats

The lecture recording system is in use since summer 2003 at the University of Osnabrück and at the University of Applied Sciences Osnabrück. During this time users with different backgrounds, knowledge and expectations experienced the system in every day use.

In fact, SVG proved to be a powerful media format in realizing the interactive navigation features described above. The problem with SVG is that it requires special viewer plugins and features like animation or interactivity are not supported by many viewers or plugin implementations differ from browser to browser and from operating systems to operating system. This has certainly lead to acceptance problems in the past. This means that the full functionality is only given in Microsoft's Internet Explorer and users who prefer other browsers like Mozilla's Firefox or Apple's Safari tended to complain.

To overcome this drawback, a cross browser version was implemented and tested. This version relied on Adobe's Flash for the communication between SVG, video and different browser components. This was a very promising way in the beginning, but users had to install a further browser plugin as well. Using many different plugins for a user interface yields to problems. One issue is that users have to select, download and install the right plugin versions for their browser environment. In the case of the current virtPresenter version users have to install Adobe´s SVG viewer and for the video playback the Real or the Quicktime player plugin. Despite specific details and system specifications, a lot of user problems concerning wrong browser settings and browser security alerts had to be handled in the past. Reducing browser plugin dependencies is an important step to enhance user acceptance. For this, a Flash only version seems to be a very promising way.

Flash especially Flashvideo is getting more and more important and widespread in the internet. The Flash plugin is small in size and available for many browsers and operating systems. In combination with Adobe´s new Flex 2 technology it is possible to implement a cross browser system solution that relies on standard Flex components for user interaction [1]. Thereby the implemented lecture recording components can be re-used for the new interface.

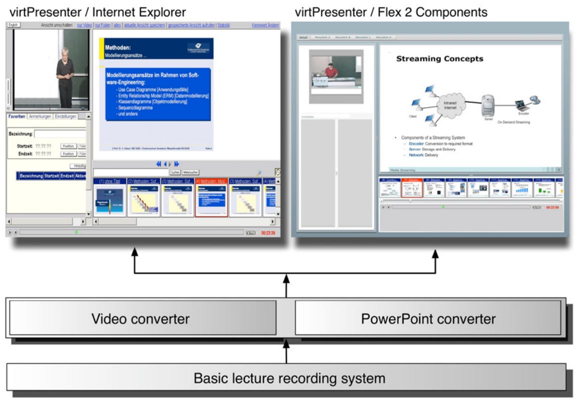

figure 6: virtPresenter Interface in IE and Flex 2 concept

Figure 6 depicts the virtPresenter user interface in Microsofts IE (left side) and the new Flex 2 Interface concept on the right side.

The basic lecture recording system is equal for both interfaces. The video converter generates a Real or Quicktime videofile for the left case and Flashvideo for the right case. The PowerPoint converter currently produces SVG and will be extended to produce Flash slides from the lecture recordings.

6. Conclusion and Further Work

Interactive content overviews are a powerful navigation aid in lecture recordings. This paper has described an SVG based implementation of such overviews and it has shown the potential of Adobe Flash as an alternative media format.

A Flash based interface version is currently being conceptualized in order to solve acceptance problems that arose from browser and plugin limitations of the SVG based version of the interface.

7. References

[1] Adobe Systems Incorporated, “Flex 2 technical overview”, technical whitepaper, 2006.

[2] Brotherton, J.A. and Abowd, G. D., “Lessons Learned From eClass: Assessing Automated Capture and Access in the Classroom”, ACM Transactions on Computer-Human Interaction, Vol. 11, No. 2, June 2004, pp. 121–155.

[3] Chen, M. “Design of a Virtual Auditorium”, Proc. ACM Multimedia 2001, Ottawa, September 2001, pp. 19-28.

[4] Day, J., Foley, J., Groeneweg, J. and Van der Mast, C., "Enhancing the Classroom Learning Experience with Web Lectures", International Conference on Computers in education, Asia-Pacific Society for Computers in Education, Singapore, December 2005, pp. 638-641.

[5] Foertsch, J., Moses, G., Strikwerda, J., & Litzkow, M. (2002). Reversing the lecture/homework paradigm using eTEACH web-based streaming video software. In: Journal of Engineering Education, 91 (3), 267-274.

[6] Graca Pimentel, M., Abowd, G.,D. and Ishiguro, Y., “Linking by interacting: a paradigm for authoring hypertext”, ACM Hypertext 2000, pp. 39-48.

[7] Grossmann, M. and Küchlin, W. “MIN -- Multimediale Module für Mathematik in Informatik und Naturwissenschaften“, Proc. DeLFI 2005, Lectures Notes in Informatics (LNI), Gesellschaft für Informatik, September 2005, pp. 201-212.

[8] Hurson, A.R. and Kavi, K.M., "Interactive teaching practices in small class sizes while cutting into the high cost of education", Proc. of the International Conference on Engineering Education, Ostrava, Czech Republic, Aug. 10-12, 1999.

[9] Hürst, W., Lauer, T. and Götz, G., “An Elastic Audio Slider for Interactive Speech Skimming", NordiCHI, October 2004.

[10] Hürst, W. and Jarvers, P., “Interactive, Dynamic Video Browsing with the ZoomSlider Interface”, IEEE ICME 2005, Amsterdam, The Netherlands, pp. 558 – 561.

[11] Lauer, T. & Ottmann, T. “Means and Methods in Automatic Courseware Production: Experience and Technical Challenges”. World Conference on E-Learning in Corp., Govt., Health, & Higher Ed. 2002(1), pp. 553-560.

[12] Lauer, T. and Trahasch, S. "Begriffsbesprechung: Vorlesungsaufzeichnungen" i-com, 2005, 3, pp. 61.

[13] Mertens, R., Brusilovsky, P., Ishchenko, S. and Vornberger, O. “Time and Structure Based Navigation in Web Lectures: Bridging a Dual Media Gap”, World Conference on E-Learning in Corp., Govt., Health, & Higher Ed. 2006. to appear.

[14] Mertens, R., Knaden, A., Krüger, A. and Vornberger, O., “Einsatz von Vorlesungsaufzeichnungen im regulären Universitätsbetrieb“. Workshop "Elektronische Unterstützung der Präsenzlehre", Ulm, Germany, September 2004, pp. 429-433.

[15] Ponceleon D. and Dieberger, A. "Hierarchical Brushing in a Collection of Video Data", HICSS'34 (Hawaii International Conference on Systems Science), Maui, HI, January 2001. pp. 1654 -1661.

[16] Smeaton, A.F. and Keogh, G.,“An analysis of the use of virtual delivery of undergraduate lectures”, Computers and Education 32, pp. 83-94.

[17] Zupancic, B. and Horz, H. “Lecture Recording and its Use in a Traditional University Course”, ACM Conference on Innovation and Technology in Computer Science, Aarhus, Denmark, 2002. pp. 24-28.