

The virtPresenter lecture recording system: Automated production of web lectures with interactive content overviews

Robert Mertens*, Markus Ketterl* and Oliver Vornberger†

*Virtual Teaching Support Center, University Osnabrück, Schloßstraße 9, 49069 Osnabrück, Germany
Email: {rmertens, mketterl}@uni-osnabrueck.de

†Dept. of Computer Science, University Osnabrück, Albrechtstraße 28, 49069 Osnabrück, Germany
Email: oliver@uni-osnabrueck.de

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 article presents the virtPresenter lecture recording system that tackles navigation in web lectures with a hypermedia navigation concept that is improved with interactive content overviews. Apart from navigation in web lectures the article also addresses didactic scenarios for web lectures and issues related to the workflow of recording lectures.

Keywords: Lecture Recording, E-Lectures, Web Lectures, Multimedia Navigation, Automated Production, SVG, Flex 2

1. INTRODUCTION

Recording live lectures has proven to be a fast and efficient way of creating content for e-learning (Lauer & Ottmann, 2002). This content can be used in a number of ways ranging from simply augmenting a lecture to profoundly changing its didactic setting. See (Mertens *et al.*, 2004 a) for an overview.

A number of application scenarios aim for replacing the lecture in its traditional form by more interactive learning sessions (Day *et al.*, 2005; Foertsch *et al.*, 2002; Hurson & Kavi, 1999). 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 takes place 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 required prior knowledge.

To fully exploit the possibilities given by electronic lectures, advanced navigation features are crucial (Lauer & Trahasch, 2005). Students must be

able to skip unnecessary passages of a recording and to repeat problematic sections. This statement does not only hold true for novel application scenarios but also when lecture recordings are used as an addition to a conventional lecture.

The virtPresenter system tackles the problem of navigation in web lectures with a hypermedia navigation concept that is improved by using interactive content overviews for both time based and structure based navigation. The article describes details of the virtPresenter and gives background information on application scenarios for lecture recordings. It also shows how overhead related to the production of web lectures can be reduced by automation of recording related workflows.

The article is organized as follows: Section 2 presents a brief analysis on how navigation in web lectures affects learning efficiency. Section 3 describes a number of application scenarios for web lectures with a special focus on whether and how the benefits of web lectures can be exploited in the scenarios described. Section 4 gives an overview of related work in the direction of interactive content overviews in lecture recordings. Section 5 describes the automated workflow of the virtPresenter lecture recording system. Section 6 discusses implementation details and section 7 shows the potential of an alternative Flash based solution. Section 8 briefly summarizes the work presented in the article.

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 (Chen, 2001). This is, however, rarely the case in traditional lectures.

In contrast to classic lectures, web lectures bring the advantage that the material presented in the lecture can be broken up into small pieces that can be replayed as often as required (Grossmann & Küchlin, 2005).

This fact and the possibility to skip known knowledge are key advantages of eLearning (Hurson & Kavi, 1999). The comparison in Figure 1 schematically depicts the impact of skipping known parts and directly navigating to relevant parts of a recording 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.

Experiences described in (Smeaton & Keogh, 1999) and (Foertsch *et al.*, 2005) as well as a student

survey published in (Day *et al.*, 2005) back this hypothesis for scenarios in which recorded lectures are used to replace the live lecture. (Zupancic & Horz, 2002) describe a study of viewing behaviour in which recordings were used as a supplement to a live lecture. Results from this study indicate that navigation plays an important role in this kind of scenarios, too.

3. APPLICATION SCENARIOS FOR WEB LECTURES

The degree to which navigation in web lectures can be exploited to improve learning efficiency depends highly on the way, in which the recordings are used for teaching. Possible application scenarios for lecture recordings in non-distance universities can be arranged in the following categories: supplement to a traditional lecture, synchronous surrogate for a traditional lecture, asynchronous replacement for a traditional lecture, non-lecture collection of learning content (Mertens *et al.*, 2004a). This section briefly characterises each of these four archetypes and outlines in how far the benefits of advanced navigation described in the previous section occur in the each scenario respectively.

3.1 Supplement to Traditional Lecture

The use of web lectures as a supplement to a traditional lecture is likely the most popular form in which web lectures are employed at non-distance university courses. In this scenario the lecture is recorded in each session and later made available on a web server or a DVD. Since the lecture takes place as usual, technical problems resulting in missing recordings are not critical. This scenario is therefore especially suitable for first-time users. The recordings can be used by students to complete their homework and to prepare for exams (Zupancic & Horz, 2002). They can also be used by students who have not attended the live lecture. Students who use the recordings to watch a session they have not attended might want to skip parts of it and repeat other parts. Especially skipping parts of the recording requires navigation facilities that rise above the functionality of simple time based navigation since the beginning of the next thematic section has to be identified. Using the recording for homework or to prepare for exams demands even more of the navigation interface, since students are searching specific information and need to browse at a fine granularity to locate answers to specific questions (Hürst, 2001).

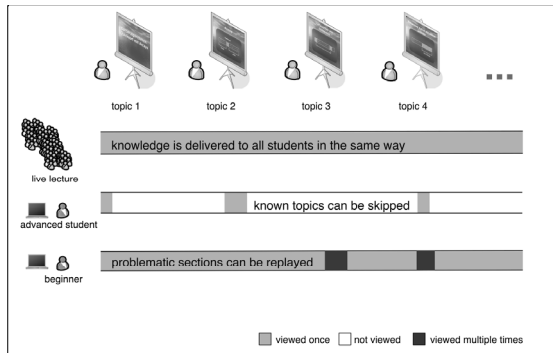


Figure 1 Schematic comparison of knowledge reception in traditional and web lectures

3.2 Synchronous Surrogate for Traditional Lecture

In this scenario, the lecturer's talk is replaced by a recorded lecture that is shown to the audience in a lecture theatre. A typical setting involves two projectors, one showing the video of the lecturer and the other one displaying the lecture slides. This scenario is aimed at situations in which schedules make it impossible for students to attend a lecture and an alternative has to be offered at another time slot. It is also applicable to situations in which the lecturer is not available because of a sabbatical or similar reasons. Since replay takes place in a lecture theatre, students do not need to use a computer. This fact makes the scenario applicable for students who are not used to technology, too. The downside of this scenario is that the benefits e-learning grants to learning efficiency do not become effective since students can not use the learning content in a self-paced fashion. This changes, however, if this scenario is combined with the scenario described in the next subsection.

3.3 Asynchronous Replacement for Traditional Lecture

The difference between this scenario and the one described in the previous subsection is that the recording can be watched at single computers and in individual sessions. Students can thus fully take advantage of the recorded material in that they can decide when to watch the lecture and how to watch it. They can repeat single passages as often as desired and they can skip arbitrary parts of the recording. Hence, navigation functionalities in the viewer interface play an important role. The importance of this role is even increased by the fact that assignments are a crucial part of this scenario and

that the recordings will usually be solved with the recording. If the scenario is implemented without periodic assignments, the students' motivation can decrease rapidly. Also, students can be led to the false assumption that learning can be postponed until shortly before the exam (Bell *et al.*, 2001). If offered in combination with tutorial support and periodic assignments, the scenario has been reported successful in a number of instances (Foertsch *et al.*, 2002; Stringer, 2003; Smeaton & Keogh, 1999).

3.4 Non-Lecture Collection of Learning Content

Recorded lectures need not necessarily be used in the context of a curricular lecture. They can also be used as sources of knowledge in other situations. Alumni can use these resources at work and students can use them for projects. Since the information relevant to these tasks can usually be found in a tiny part of the recording, search and navigation are essential (He *et al.*, 2000). Another use of web lectures outside a curricular lecture context is preparing for exams at the end of studies. In this case, students need to skip irrelevant parts of the recording and need to be able to find relevant passages quickly.

4. RELATED WORK

Even though lecture recording can in principle be realised for a number of different presentation formats such as chalkboard lectures, application demos, or electronic slide presentations (Liu & Kender, 2004), virtPresenter is designed for recording PowerPoint lectures only. This decision was driven by the fact that the main goal behind the development of virtPresenter is navigation in web lectures. PowerPoint allows for tracking events such as slide changes and slide animation steps. Also, textual information contained on slides can easily be extracted and associated with the PowerPoint event it belongs to.

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. The most basic and most commonly found navigation aids are timelines and slide overviews. Figure 2 shows these two elements as used in the virtPresenter viewer interface.

Timeline and slide overview constitute inherently different navigation tools. While the timeline gives a temporal overview, the slide overview pro-

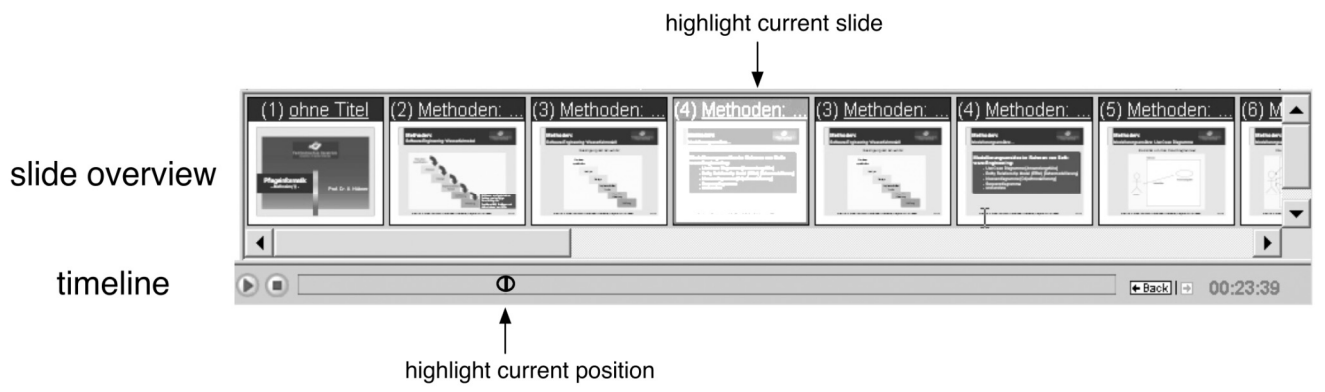


Figure 2 Timeline and slide overview in the virtPresenter web lecture interface

vides for orientation in the structure of the recording. Both the time based and the structure based navigation paradigm have been subject to a number of research efforts with the aim to improve navigation in web lectures. This section gives an overview of recent developments in these two paradigms with a special focus on how interactivity has been employed to facilitate navigation in the approaches presented.

4.1 Advanced Time Based Navigation

During the last few years, a number of approaches have emerged that improve the classical slider based navigation described above. These approaches are random visual scrolling, hierarchical brushing, elastic panning and the Zoomslider.

The common idea behind these approaches is a preview functionality that is triggered by moving the slider knob along the timeline or by brushing over the timeline using the mouse pointer. This allows for a fast assessment of the lectures content that is similar to flipping through the pages of a book (Hürst & Müller, 1999). Hierarchical brushing and the Zoomslider interface augment this functionality by adding possibilities to adjust the resolution of the timeline. In hierarchical brushing the timeline is divided into segments and brushing over a segment opens a number of corresponding sub-segments (Ponceleon & Dieberger, 2001). This process can be realised recursively so that very high degrees of resolution can be achieved.

In the Authoring on the Fly system (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 (Lauer & Ottmann, 2002). 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 like elastic panning (Hürst & Götz, 2004) and the Zoomslider (Hürst & Jarvers, 2005).

4.2 Advanced Structure Based Navigation

A number of web lecture interfaces feature different advanced structure based navigation aids. As all of these navigation aids can be classified as hypermedia navigation features, this section will briefly introduce hypermedia navigation in web lectures. A number of systems implement one or two hypermedia navigation features. However, virtPresenter is the only web lecture system integrating all standard hypermedia navigation features in one concept. An overview over the distribution of hypermedia navigation features in web lecture systems can be found in (Mertens *et al.*, 2004b; Mertens *et al.*, 2006a).

Navigation in classical text or picture based hypermedia comprises five different navigation functionalities. These are footprints, bookmarks, full text search, backtracking and structural elements (Bieber, 2000). The hypermedia navigation concept for lecture recordings introduced in (Mertens *et al.*, 2004b) features all of these functionalities in a form that is altered to accommodate for the time based nature of the medium. Footprints are realised as marks on the timeline that highlight visited passages. Bookmarks are implemented as a time based tool so that arbitrary passages can be bookmarked. Full text search indexes all words on the slides that are used during the lecture. Backtracking gives users the possibility to undo navigation actions that lead to an undesired target by jumping back to the replay position that was active before the respective navigation act was

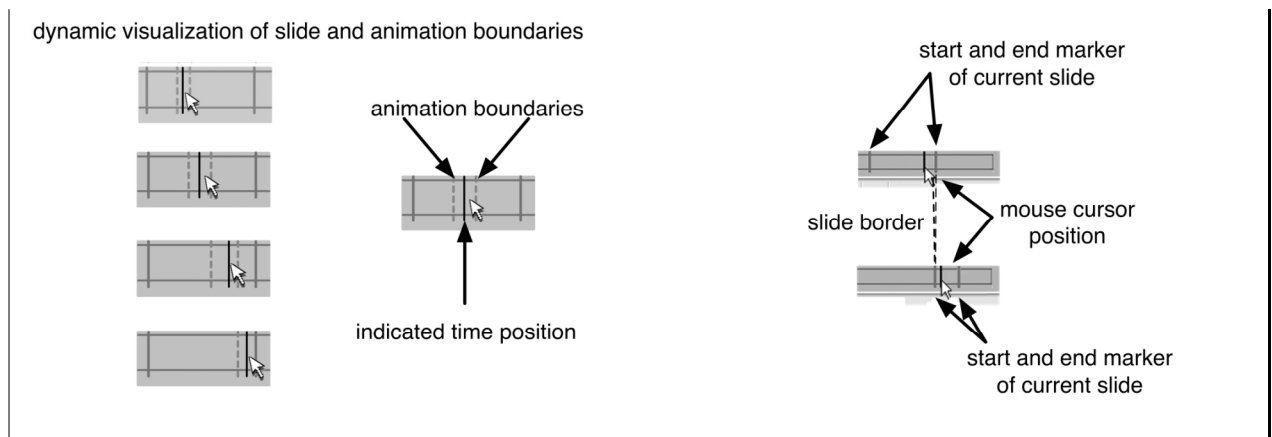


Figure 3 Dynamic visualization of slide and animation boundaries

performed. Structural elements can be navigation elements in the shape of arrows that allow jumping to the start of the previous or next animation and slide. A more advanced form of structural elements are interactive slide overviews.

4.3 Interactivity

The approaches for advanced time based navigation described above already make use of interactivity in that they offer a dynamic preview for arbitrary positions of the timeline. In the ePresence system¹ the timeline is used in full text search to highlight starting positions of slides that contain a search string. eClass (Graca Pimentel *et al.*, 2000) introduces an approach called linking by interacting in which handwritten annotations that are added to a slide during the lecture are embedded 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.

The next section shows further improvements to navigation that can be realized with interactive content overviews for both structure and time based navigation.

5. IMPROVING NAVIGATION WITH INTERACTIVE CONTENT OVERVIEWS

As described above, many lecture recording systems feature both time- and structure based overview components, e.g. a timeline and a thumbnail overview or a table of contents. This section describes how the timeline and the thumbnail overview of virtPresenter have been enriched with interactive functionality in order to facilitate navigation.

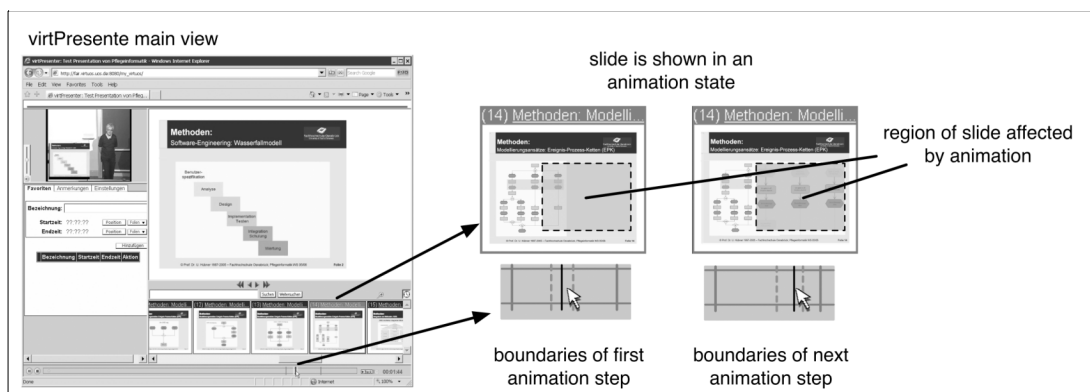


Figure 4 Navigation in virtPresenter

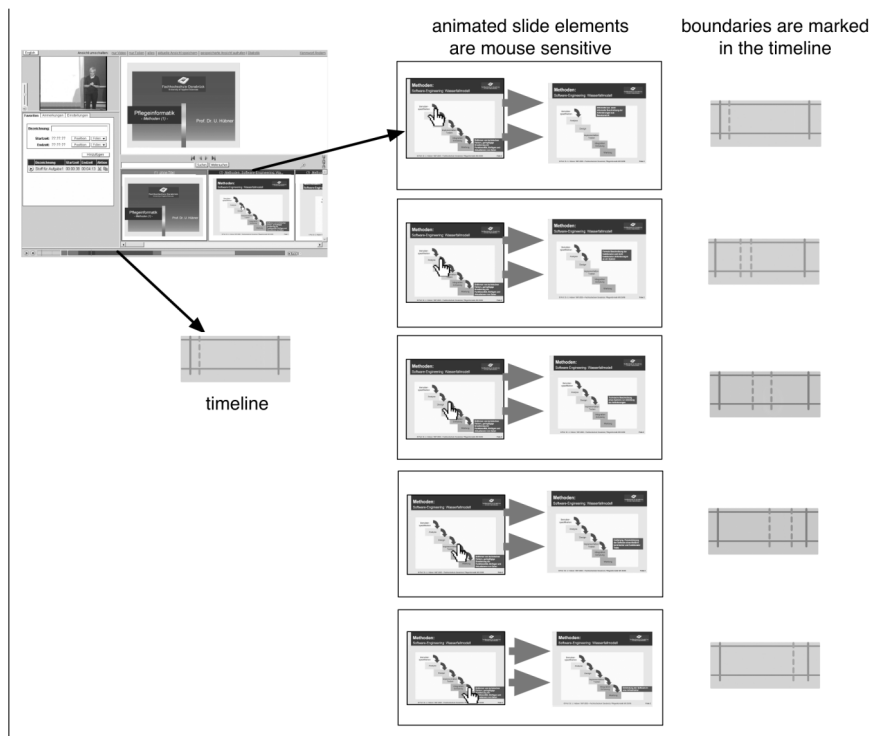


Figure 5 Interactive slide elements

5.1 Time Based Navigation

In virtPresenter, time based navigation using the timeline is improved by visualizing structural context information belonging to the current slider position. Structural contexts can be the slides shown at the indicated time position or slide animation steps that belong to the indicated time position. The visualization is realised by boundary markers that highlight the time positions when the slide or animation step is shown and the time position of the next animation step or slide change. Adding this information at the level of slide animation steps like items of a bullet list that appear one after the other brings the advantage of intra slide granularity. Visualizing all boundaries of slide changes and animation steps does, however, clutter the timeline with a huge amount of information that is at most times unused and even distracting.

In the virtPresenter viewer interface, this information is therefore 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 3. This way, the adjacent animation steps or slides can easily be located. 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. Figure 4 shows how the connection of these two approaches is realized in the virtPresenter interface.

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 (see Mertens *et al.*, 2004b for a more detailed description of this feature).

5.2 Structure Based Navigation

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.

Text Elements can 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 5.

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 (Brotherton & Abowd, 2004). 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 (Graca Pimentel *et al.*, 2000). In virtPresenter, PowerPoint slides are converted to SVG using the SVGmaker software.² During this process, information about which element of a slide was animated is lost. The virtPresenter uses an approach that identifies the animated elements from a PowerPoint slide in an SVG document generated from this slide. Generating SVG slides from PowerPoint is part of a fully automated production chain described in the next section.

6. FROM LIVE LECTURE TO WEB LECTURE

The process of creating a web lecture from a live

lecture can be divided into four steps which are preparation phase, recording phase, post-processing phase and usage phase (Hürst *et al.*, 2004 b).

In the preparation phase slides are imported by the recording software used.

In the recording phase the presentation is recorded in the classroom. This phase usually involves setting up recording equipment like camera and microphones as well dismantling it afterwards.

In the post-processing phase, the recorded material is transformed into the actual web lecture. This step involves synchronising the recorded video with the slides used in the lecture. In most systems this process also includes transforming the slides into a different format for replay. After this conversion is finished the result has to be moved to a server and a link to it has to be posted.

The usage phase does usually not involve any work related to technical aspects.

Since especially set-up and dismantling of recording equipment requires a huge amount of manual work, the workflow of web lecture production with the virtPresenter system is highly automated (Mertens *et al.*, 2005). A major point in this automation is making the distinctions between the above mentioned phases transparent to the user. Accordingly, the recording phase is started automatically after the fully-automated preparation

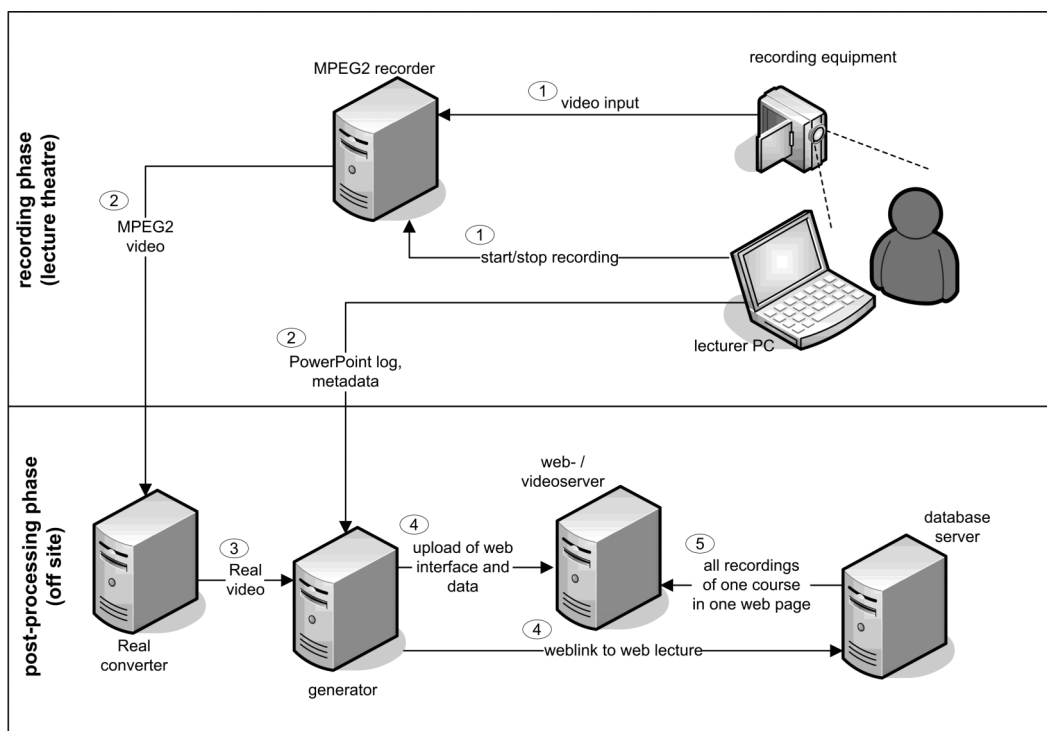


Figure 6 Automated workflow in recording and post-processing phase

phase and when the lecturer finishes the recording, post-processing is started in the background.

6.1 Automated Workflow

Figure 6 gives an overview of all processing steps in the virtPresenter production chain. As the preparation phase is integrated into the recording phase and since the usage phase does not have any implications for automation, the figure focuses on the recording and post-processing phase only. The upper half of the figure shows all steps in the recording phase and the lower half shows all steps in the post-processing phase.

The automated workflow is depicted counter-clockwise starting in the upper right half of figure 6. In the first step (1) recording on the MPEG 2 encoding computer is started and PowerPoint is started with the presentation selected for recording. Until the PowerPoint presentation is ended by the lecturer, slide change events and animation events from PowerPoint are logged in a file. When the presentation is finished, a stop signal is sent to the MPEG 2 encoding computer (1). Subsequently the MPEG 2 video is transferred to a server (2) where it is converted into Real video.

Simultaneously the PowerPoint file used in the presentation, the event log file and course metadata is moved (2) to a central server called Generator in the figure. After the MPEG 2 video is converted to Real it is also uploaded to the Generator (3). The Generator transforms the incoming PowerPoint file into SVG and builds the web interface using the SVG slides, the event log file and the Real video. When the web interface is completed it is uploaded on a web server (4). The Real video file is uploaded to server that allows for streaming. However, for the end users, this separation of video and interface is transparent as the interface displays both the

slide and the video. Finally a link to the interface is automatically stored in a database together with course data (4). This way a page containing links to all recorded lectures from a specific course is generated automatically (5).

The MPEG 2 format was chosen for video because it can be encoded using optimized hardware and it allows for recording in DVD quality. Since MPEG 2 is, however, not suited for web delivery, the video is later converted to the Real video format. The fact that both Real Video and MPEG 2 use lossy data compression does not result in inferior quality since the MPEG 2 encoding is realised in DVD quality and the Real Video is encoded in lower quality.

6.2 SVG Processing

The thumbnail slides in the slide overview are in SVG. This format was chosen because it is a vector graphics format based on XML. Being a vector graphics format, SVG stores data such as text or geometrical forms in a symbolic way. The fact that it is based on XML facilitates processing with SVG documents. For display in a web browser, SVG can be used like any other graphics format, provided it is supported by the browser or an additional plug in such as the Adobe SVG Viewer is installed.

To construct the web interface the Generator generates one SVG slide for each animation step in the PowerPoint presentation. The underlying algorithm is realized in two steps. The first step is sequencing a slide in a way that a static image of each animation is stored on a PowerPoint slide. This is realized by a custom program that uses PowerPoint's COM interface. In the second step these intermediate slides are printed as SVG files using SVGmaker. The slides shown in the slide overview component of the interface are subjected to a number of post processing operations. The most important of these are indexing for full text search and adding scripting to animated shapes on the slides. Indexing for full text search is realized as a rather simple procedure as SVG already contains text in text-tags. The algorithms needed to add script to animated elements of a slide require, however, more complex computing. 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 (called reference document in the further course of this article) by the same custom program that is used for sequencing animated slides.

In a subsequent step, the content of each SVG document is compared to the content of the SVG

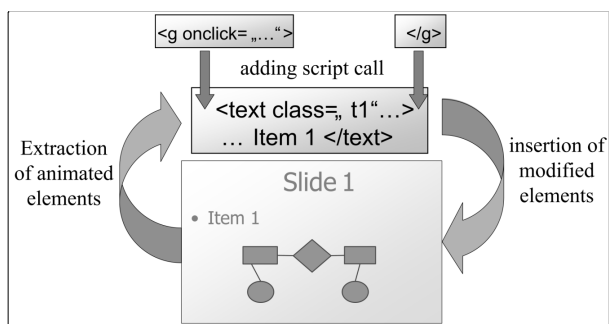


Figure 7 Script code is added to existing SVG

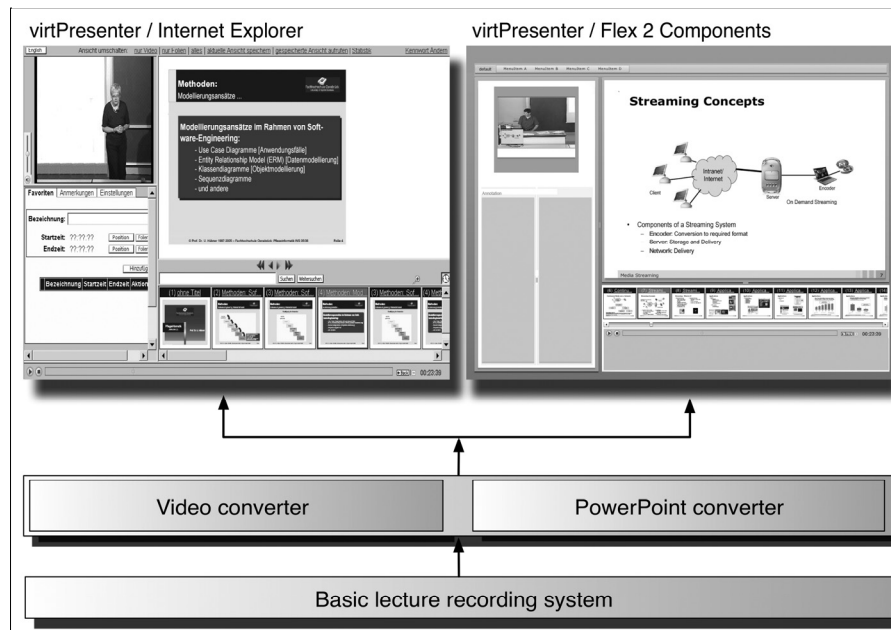


Figure 8 VirtPresenter Interface in IE and Flex 2 concept

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 group node that contains the script code for this element as shown in Figure 7.

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. This requires a number of steps that are described in greater detail in (Mertens *et al.*, 2006 b).

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

Once the SVG used in the slide overview is fully processed, a JavaScript file containing all synchronization data is created. Together with a folder of all SVG slides this file is inserted into a template for the viewer interface.

7. FUTURE TRENDS

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

In this time, SVG proved to be a powerful media format in realizing the interactive navigation features described above. The main problem with SVG is that it requires special viewer plug-ins. Also, features like animation or interactivity are not supported by many viewers. Another problem is constituted by the fact plug-in implementations differ from browser to browser and from operating system to operating system. As a result, the full functionality of the interface is only given in Microsoft's Internet Explorer. Hence 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 Flash for the communication between SVG, video and different browser components. From a technical perspective, the solution was very promising, but users had to install yet another browser plug-in for Flash. Using many different plug-ins for a user interface, however, can lead to a number of problems. One issue is that users have to select, download and install the right plug-in versions for their browser environment. In the case of the current virtPresenter version users have to install Adobe's SVG viewer and a plug-in for video playback (Real Player or QuickTime, depending on which virtPresenter version is used). Because of the complexity of installing the right version of the right plug-in, user problems concerning wrong browser settings and browser security alerts had been very common in the past. Another problem with plug-ins is that changes brought by versions to come can not be foreseen at the time the interface

code is written. In a worst case scenario, vendors can even stop maintaining their product and users can no longer download the plug-ins. Reducing browser plug-in dependencies is therefore an important step to enhance user acceptance. For this, a Flash only version seems to be a very promising way.

Flash is a platform for delivering many types of media including animation and interactive content for web pages and user interfaces. The Flash plug-in is small in size and available for many browsers and operating systems. The new Flash 9 Player can be downloaded for Windows, Apple and even Linux systems. A big advantage is that many users already have the Flash plug-in installed on their computer. Flash is very similar to SVG in that it is vector based, allows for hyper linking, can be animated, zoomed and panned, offers database connectivity and is scriptable (House & Pearlman, 2002).

In combination with Adobe's new Flex 2 technology, Flash can be used to implement a cross browser system solution that relies on standard Flex components for user interaction (Adobe, 2006). Flex is a rich Internet application framework based on Flash that enables us to re-use already implemented logic components for a new cross-browser Flash interface.

Figure 8 depicts the virtPresenter user interface in Microsoft's IE (left hand side) and the new Flex 2 interface concept on the right hand side.

The basic lecture recording system is the same for both interfaces. The video converter generates a Real or QuickTime video file for the current IE interface version and Flash video (a special video standard that is designed to be played in Flash applications) for the Flex 2 interface. The PowerPoint converter currently produces SVG and will be extended to produce Flash slides from the presentations used for recording. Most of the existing code and concepts can easily be adapted to a workflow producing recorded lectures with the Flash interface.

8. CONCLUSION

In most application scenarios for recorded lectures, navigation plays a crucial role. This article has given an overview of different application scenarios for recorded lectures and discussed the state of the art of navigation in web lectures. It has also shown how navigation in web lectures can be improved with interactive content overviews. The virtPresenter web lecture system featuring interactive content overviews and a fully automated production chain has been introduced and problems with the currently used media format SVG have been discussed. As a solution to these problems a concept for real-

ising the functionalities implemented in the current virtPresenter web interface with Adobe's Flex 2 has been presented.

REFERENCES

- Adobe Systems Incorporated (2006) Flex 2 technical overview, technical whitepaper, Available at http://www.adobe.com/products/flex/whitepapers/pdfs/flx2wp_technicaloverview.pdf (retrieved October 2006).
- Bell, T., Cockburn, A., McKenzie, B. and Vargo, J. (2001) Flexible Delivery Damaging to Learning? Lessons from the Canterbury Digital Lectures Project. *ED-MEDIA 2001: Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Association for the Advancement of Computing in Education, Tampere, Finland, pp. 117-122.
- Bieber, M. (2000) Hypertext. *Encyclopedia of Computer Science (4th Edition)*, Ralston, A., Edwin Reilly and David Hemmendinger (eds.), Nature Publishing Group, pp.799-805.
- Brotherton, J.A. and Abowd, G. D., (2004) Lessons Learned From eClass: Assessing Automated Capture and Access in the Classroom, *ACM Transactions on Computer-Human Interaction*, Vol. 11, No. 2, pp. 121-155.
- Chen, M. (2001) Design of a Virtual Auditorium", *Proc. ACM Multimedia 2001*, Ottawa, pp. 19-28.
- Day, J., Foley, J., Groeneweg, J. and Van der Mast, C., (2005) Enhancing the Classroom Learning Experience with Web Lectures, *International Conference on Computers in education*, Asia-Pacific Society for Computers in Education, Singapore, pp. 638-641.
- Foertsch, J., Moses, G., Strikwerda, J. and Litzkow, M. (2002) Reversing the lecture/homework paradigm using eTEACH web-based streaming video software. In: *Journal of Engineering Education*, 91 (3), pp. 267-274.
- Graca Pimentel, M., Abowd, G.,D. and Ishiguro, Y., (2000) Linking by interacting: a paradigm for authoring hypertext, *ACM Hypertext 2000*, pp. 39-48.
- Grossmann, M. and Küchlin, W. (2005) MIN — Multimediale Module für Mathematik in Informatik und Naturwissenschaften, *Proc. DeLFI 2005, Lectures Notes in Informatics (LNI)*, Gesellschaft für Informatik, pp. 201-212.
- He, L.; Grudin, J.; Gupta, A. (2000) Designing presentations for on-demand viewing. In: *ACM 2000 Conference on Computer supported cooperative work*, pp. 127-134.
- House, L. and Pearlman, E. (2002) SVG for Web Developers. Series: Advanced Web Development Series. ISBN-13: 978-0-13-100499-3; Published: Dec 17, 2002.
- Hurson, A.R. and Kavi, K.M., (1999) 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, Available at <http://www.fs.vsb.cz/akce/1999/ICEE99/Proceedings> (retrieved October 2006).
- Hürst, W. (2001) Browsing and navigation of AOF documents as an example for easy and comfortable interaction with general multimedia documents, D.W. Fellner, N. Fuhr, I. Witten (Workshop Chairs): *ECDL Workshop "Generalized Documents"*, Darmstadt, Germany, Available at ad.informatik.uni-freiburg.de/~huerst/Publications/2001_ecdl-v3d2.pdf

- (retrieved November 2006).
- Hürst, W. and Müller, R. (1999) A synchronization model for recorded presentations and its relevance for information retrieval. *Seventh ACM international conference on Multimedia*. Orlando, Florida, US, pp.333–342.
- Hürst, W., Lauer, T. and Götz, G., (2004 a) An Elastic Audio Slider for Interactive Speech Skimming, *NordiCHI*, Tampere, Finland, pp. 277–280.
- Hürst, W., Mueller, R., and Ottmann, T., (2004 b) The AOF Method for Production, Use, and Management of Instructional Media. In *Proceedings of ICCE 2004, International Conference on Computers in Education*, Melbourne, Australia.
- Hürst, W. and Jarvers, P., (2005) Interactive, Dynamic Video Browsing with the ZoomSlider Interface *IEEE ICME 2005*, Amsterdam, The Netherlands, pp.558–561.
- Lauer, T. and Ottmann, T. (2002) 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.
- Lauer, T. and Trahasch, S. (2005) Begriffsbesprechung: Vorlesungsaufzeichnungen. *i-com*, 4, 3, 2005, pp. 61.
- Liu, T. and Kender, J. (2004) Lecture Videos for E-learning: Current Research and Challenges. In *Proceedings of IEEE International Workshop on Multimedia Content-based Analysis and Retrieval*, pp 574–578.
- Mertens, R., Brusilovsky, P., Ishchenko, S. and Vornberger, O. (2006 a) 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, pp. 2929–2936.
- Mertens, R., Ketterl, M. and Vornberger O. (2006 b) Interactive Content Overviews for Lecture Recordings. *Eighth IEEE International Symposium on Multimedia (ISM'06)*, pp. 933–937.
- Mertens, R., Knaden, A., Thelen, T. and Vornberger, O. (2005) Kopplung von LMS und Vorlesungsaufzeichnungssystemen: Voraussetzungen und Potentiale. In: Horz, H., Hürst, W., Ottmann, T., Rensing, C & Trahasch, S. eLectures-Einsatzmöglichkeiten, Herausforderungen und Forschungsperspektiven. In: Lucke, U., Nölting, K., Tavangarian, D. (eds.) *Workshop Proceedings DeLFI 2005 and GMW 2005*. Rostock, pp. 43–48.
- Mertens, R., Knaden, A., Krüger, A. and Vornberger, O., (2004 a) Einsatz von Vorlesungsaufzeichnungen im regulären Universitätsbetrieb. *Workshop "Elektronische Unterstützung der Präsenzlehre"*, Ulm, Germany, pp. 429–433.
- Mertens, R., Schneider, H., Müller, O. and Vornberger, O. (2004 b) Hypermedia Navigation Concepts for Lecture Recordings. *E-Learn 2004: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Washington, DC, USA, pp. 2480–2487.
- Ponceleon D. and Dieberger, A. (2001) Hierarchical Brushing in a Collection of Video Data, *HICSS'34 (Hawaii International Conference on Systems Science)*, Maui, HI, pp. 1654–1661.
- Smeaton, A.F. and Keogh, G. (1999) An analysis of the use of virtual delivery of undergraduate lectures, *Computers and Education* 32, pp. 83–94.
- Stringer, K.S. (2003) Video technologies for Database course delivery, *Proceedings of Teaching, Learning and Assessment in Databases (TLAD)*, Coventry, LTSN, pp. 113–117.
- Zupancic, B. and Horz, H. (2002) Lecture Recording and its Use in a Traditional University Course, *ACM Conference on Innovation and Technology in Computer Science*, Aarhus, Denmark, pp. 24–28.

NOTES

1. www.epresence.tv
2. www.svgmaker.com

Robert Mertens has studied Cognitive Science at the University of Osnabrück and is currently a Ph. D. student in Computer Science. His current research interests are lecture recording systems and navigation in recorded lectures. He is working at the virtual teaching support center (virtUOS) at the University of Osnabrück. Robert has authored and co-authored a number of papers on lecture recording and other e-learning topics. His experience includes stays at the University of Pittsburgh, PA, the DaimlerChrysler Research and Technology Center, Palo Alto, CA and DaimlerChrysler, Esslingen, Germany.

Markus Ketterl has studied Computer Science at the University of Applied Sciences Osnabrück. His current research interests are object detection and recognition in video and audio, social navigation, applications for mobile devices and alternative distribution possibilities for media content. He is working at the virtual teaching support center (virtUOS) at the University of Osnabrück. Besides this he is also working as a freelancer in many media production projects (audio- and video production and editing, computer graphics, software engineering and media design) for different companies.

Oliver Vornberger, born 1951, graduated in computer science from the University of Dortmund in 1976 and worked as research and teaching assistant at the University of Paderborn. After his PH.D. in 1980 he spend a year as a postdoc at the University of California at Berkeley. Since 1989 he has a full professorship at the University of Osnabrück and is head of the multimedia working group with a focus on web publishing and computer graphics.

