# Vector Graphics for Web Lectures: Experiences with Adobe Flash 9 and SVG

**Markus Ketterl*, Robert Mertens** and Oliver Vornberger†**

***Virtual Teaching Support Center, University Osnabrück, Schloßstraße 9, 49069 Osnabrück, Germany**

**Email: mketterl@uni-osnabrueck.de**

****Fraunhofer IAIS, Schloß Birlinghoven, 53754 Sankt Augustin**

**Germany**

**Email: robert.mertens@iais.fraunhofer.de**

**† Dept. of Computer Science, University of Osnabrück, Albrechtstraße 28, 49069 Osnabrück, Germany**

**Email: oliver@uni-osnabrueck.de**

## ABSTRACT

Vector graphics are an ideal content format for slide based lecture recordings. File sizes can be kept small and graphics can be displayed in superior quality. Information about text and slide objects is stored symbolically which allows texts to be searched and objects on slides to be used interactively, e.g. for navigation purposes. The use of vector graphics for web lectures is, however, a trend that has begun only recently. A major reason for this is that multiple media formats have to be combined in order to replay video and slides. This article presents experiences made during the development and every day use of two versions of the lecture recording system virtPresenter. The first of these versions is based on SVG while the second one is based on Adobe Flex2 (Flash 9) technology.

**Keywords:** Lecture Recording, Web Lectures, SVG, Flash 9, Flex 2, Adobe AIR, Multimedia, Vector Graphics

## 1. INTRODUCTION

Vector based graphics formats offer a number of possibilities for the realization of web lecture interfaces for slide based talks. One major advantage is that they support capturing contents in a symbolic manner which is a requirement for searching text in a recording (Lauer and Ottmann 2002). They also offer superior picture quality. Last but not least, vector based graphics formats enable developers to realize a high degree of interactivity that can be used for implementing advanced navigation concepts as described in (Mertens et al. 2006 a). They also can be used to tackle a number of layout problems as further described in (Mertens et al. 2006 b).

Vector graphics are, however, not very common in web lectures. This article presents the authors´ experience with two different vector graphics formats: SVG (scalable vector graphics) and Adobe's new Flex 2 (Flash 9 based) technology for content presentation and control in the web lecture system virtPresenter.

The SVG based version of 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. The Adobe Flex 2 based counterpart has been introduced in February 2007 after a seven month development and testing period. This new

version is, apart from small changes concerning further system requirements and improvements, in productive use since March 2007.

The article is organized as follows: Section 2 points out the advantages vector graphics can bring for web lectures and briefly presents a hypermedia navigation interface for web lectures that is based on SVG. Section 3 describes experiences with this SVG based interface and points out difficulties that arose during the use of this interface in a number of university courses. Section 4 compares Flash and SVG with respect to their use in lecture recording. Section 5 introduces the Flash based successor of the SVG based interface. Section 6 describes changes in workflows for administrators and users that have become possible with Flash. Section 7 briefly summarizes the work presented in this article and refers to future projects and ideas.

## 2. ADVANTAGES OF VECTOR GRAPHICS IN WEB LECTURES

The advantages of using vector graphics for content representation in web lectures can be summarized in a couple of words: vector graphics store content in a symbolic way, vector graphics can be enlarged without loss of quality and many vector graphics formats allow for interactive on-the-fly manipulation of contents. The aim of this section is to show why these properties of vector graphics are useful by showing how each of them improves web lectures.

### Symbolic representation of contents and interactivity

The original virtPresenter user interface shown in figure 1 was developed to implement a hypermedia navigation concept for lecture recordings (Mertens 2007). Hypermedia navigation consists of the five elements full text search, bookmarks, backtracking, structural elements and footprints (Bieber 2000).

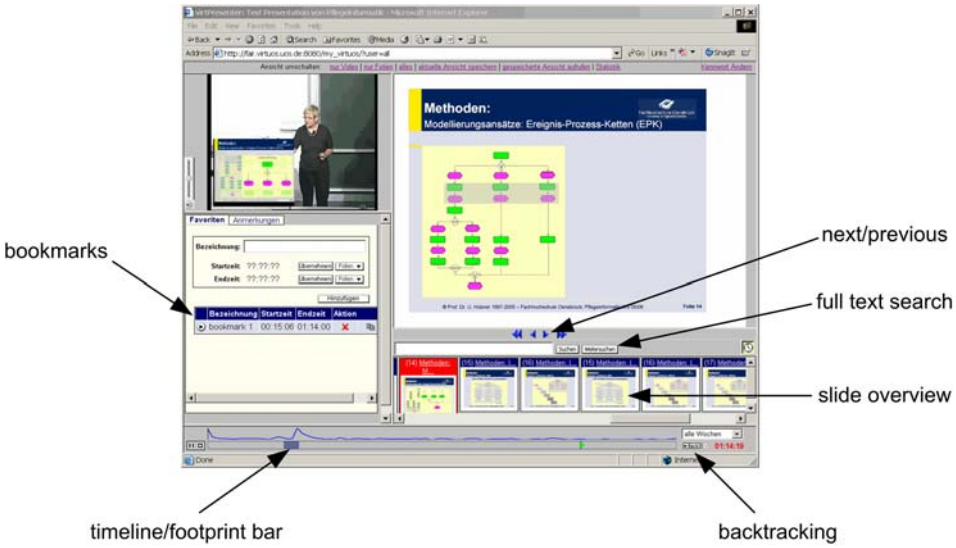


**Figure 1: virtPresenter 1.0 user interface**

Full text search is realized by searching the text of the slides in the slide overview. Search results are highlighted by an animation that grows and shrinks them repeatedly. Both the ability to search in the slides directly and to animate search results is based on the properties of SVG (symbolic representation and manipulation on the fly). Bookmarks are realized as a functionality that allows for selecting arbitrary passages and storing them for later viewing or exchanging them with other students. Backtracking is implemented by storing the play position whenever the user navigates to another play position. Thus each navigation action can be undone. In order to facilitate orientation at the stored play positions, replay begins at their time index minus three seconds. Structural elements are realized in two ways the simple one of which are next/previous buttons that allow navigating to the next or previous slide or animation step. A more sophisticated realization of structural elements is the interactive slide overview implemented in virtPresenter (Mertens et al. 2006 c). In the overview, those parts of a slide that had been animated during the original presentation when the lecture was recorded can be clicked on with the mouse. The recording then starts replay at the time index when the respective animation takes place during the

lecture. To realize these features, the slide documents are analyzed and script code containing the respective time indices is added automatically to the animated elements of a slide (Mertens et al. 2007). The implementation of this step was relatively easy due to the symbolic representation of the slide elements in SVG. Footprints serve the purpose of showing users which parts of a hyperdocument they have already visited. In classic hypertext, this is done by colouring visited and non-visited links differently. Since web lectures are time based media, another approach had to be found. In virtPresenter, coloured parts of the timeline indicate that the corresponding passages of the recording have already been watched by the user. Multiple visits are indicated by deeper shadings. The footprints are stored symbolically as pairs of start and end time indices. They are drawn on the fly when a lecture is watched. This has been realized by the use of animated SVG rectangles. The different colour shadings are created by overlapping semitransparent rectangles.

This brief description shows that the properties of SVG as a vector graphics format have been crucial for the realization of the virtPresenter user interface. Especially the implementation of footprints, bookmarks and full text search has been facilitated immensely by SVG as a vector graphics format.

**Superior picture quality**

Good picture quality of lecture slides is important even for standard usage scenarios (Ziewer and Seidl 2002). However, it becomes even more important, when the lecture slides are shown on a large screen as in the scenario depicted schematically in figure 2.

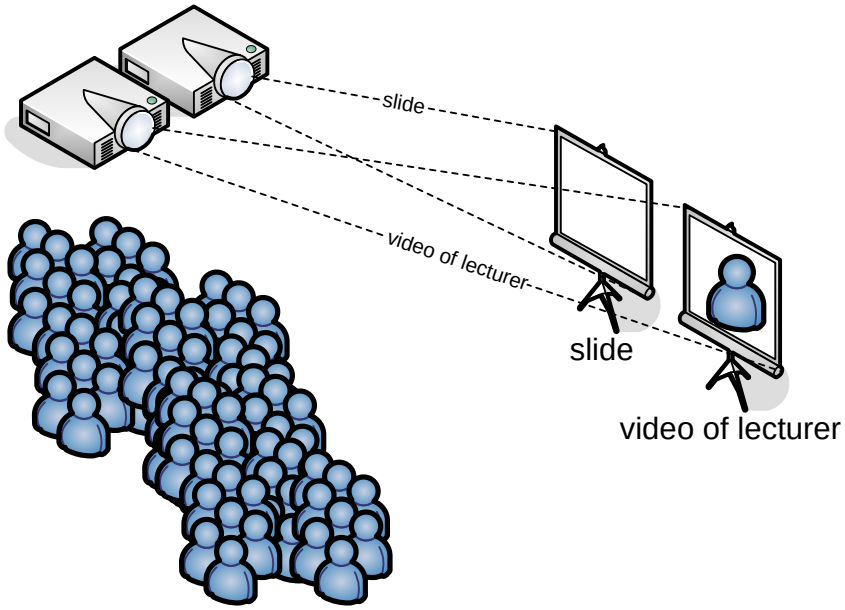


**Figure 2: Lecture slides on large screens**

In this scenario, the lecture is replaced by a cinema-like session in which the recording of the lecturer and the slides are presented to the audience on two large screens. This scenario has been carried out successfully at the University of Osnabrück a number of times (Mertens et al. 2005). Since the slides are shown on a large screen, bad picture quality becomes even more obvious than during replay on a standard computer display. At the University of Osnabrück, the slides used had been in SVG and had thus been presented in the same quality as in the original lecture.

## 3. LESSONS LEARNED

The SVG-based version of the viewer interface was first developed in 2003 and improved in various steps. The main focus of the development was to implement the hypermedia navigation concept for lecture recordings described in section 2 and in more detail in (Mertens et al. 2004).

At the time when development of the SVG based version began, SVG seemed to be a promising choice for a content format to be used in lecture recordings. SVG is an XML based vector graphics format and was expected to grow in importance. We had expected that SVG renderers supporting the required subset of the SVG standard would soon become available on more platforms than Windows and that their performance would increase in order to rival that of Macromedia Flash (now Adobe Flash). Things have, however, developed in a different direction.

While all the features described in (Mertens et al. 2004) could be realized with a combination of JavaScript, SVG and Real Video, the technology used lead to a number of problems in every-day use. Loading and rendering speed has shown to be a major problem when combining SVG and Real technology. Table 1 compares slide loading times of the SVG and the Flash based implementation (further described in sections 4 and 5). It also shows loading times for an optimized version of the SVG slides in which background graphics in the slides (logos) had been deleted to speed up rendering. The testing environment was a Windows XP system with an AMD Athlon 64 based processor with 2,01 GHz and 1 GB RAM. The tests were made locally on that system without internet connection interferences. This test indicates the elapsed time till a slide object is loaded and fully available in the main application.

| Technology | SVG | SVG optimized | Flash |
|---|---|---|---|
| average slide loading time (ms) | 164* | 120* | 67 |
| average slide loading time + video (ms) | 430** (Real video) | 243** (Real video) | 81 (Flash video) |
| average slide size (KB) | 54 | 25 | 28 |

13 different converted PowerPoint slides
* outlier here: 520, 635
** outlier here: 7300, 6349, 2280, 4300

System: Windows XP AMD Athlon 64 Processor; 2,01GHz, 1GB Ram

**Table 1: Slide loading with SVG and Flash**

As some interactivity and animation features of SVG that are only supported in the Adobe SVG Viewer (ASV) had been used in the interface, replay was only possible with the ASV for Microsoft's Internet Explorer (IE). This viewer plug-in does, however, exhibit low rendering speeds and support will be discontinued in January 2008. This fact is especially problematic when many slides have to be shown at once as it is the case for overviews. Also switching from one slide to another happens with a noticeable delay. The Real video player buffers data when users navigate in the video. This buffering also slows down the interfaces responding times noticeably. Another problem with SVG was that the plug-in required only exists for Microsoft's Internet Explorer. Even though Adobe had implemented plug-in-versions for other browsers, only the one for IE supports the subset of the SVG specification required for the implementation. This fact rules out platform independence for the interface. Last but not least, the fact that plug-ins are required for both Real Video and SVG poses an obstacle for first time users of the interface.

The use of the SVG-based interface has been evaluated in a number of courses. In these evaluations, the above mentioned points have shown to have a considerable negative impact on user acceptance. In 2006, three courses have been evaluated with a questionnaire developed for the evaluation of e-Learning at the University of Osnabrück. For abbreviation purposes, these courses are referred to in the paper as courses A, B and C. Table 2 summarizes relevant details on the courses.

| Course | Full Name of Course | Didactic Setting | Number of Students |
|---|---|---|---|
| A | Fundamentals of Biblical Theology | Lecture took place as usual, all students could attend and the recordings had been provided as an add on. | 25 |
| B | Internet Technologies | Lecture took place at one University and was transmitted to another one. Recordings were provided as an add on. A more detailed description of the scenario can be found in (Hoppe et al. 2007). | 27 |
| C | Managing Innovation and Projects | Same as course C. | 19 |

**Table 2: Course details**

Figure 3 shows how the students judged download times of the recordings. No actual download was offered. The term "download times" does thus refer to loading and rendering times of the viewer interface. By and large the numbers in the figure do not seem too critical at first sight. In practice, however, the interface loads considerably longer than other material found on the course web site. Also, the results show that while the loading times have been acceptable for most students, they have not been acceptable for all students.

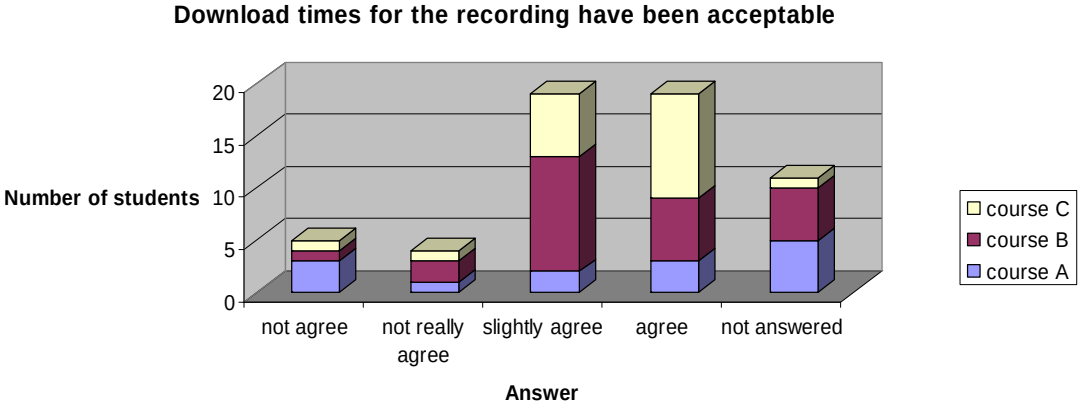


**Figure 3: Lecture recordings download times**

Figure 4 shows how many students reported problems using virtPresenter. The problem descriptions were entered as free text answers in the questionnaires. In course A, no student reported a problem. This might be due to the fact that students were given very detailed instructions. Having a non-technical background, the students have very likely followed these instructions closely. The questionnaires have also shown that all students in course A used IE. In the other courses, the questionnaires have shown that some students did not use IE (even though they had been instructed that using another web browser would cause problems with the interface). In contrast to course A, course B and C had been attended by a number of students with technical backgrounds. The questionnaires lead to the assumptions that some of these students, being used to solve problems by trial and error, have tried to use the interface with other browsers than IE unregarding the information that it would not work on these browsers. Seemingly unaware of the fact that the interface was not supported under these settings, the students reported the system behaviour as faults. From one problem description it even became clear, that the student had not installed any SVG viewer.

In order to counter the above described effects, a number of improvements had been devised for the SVG based version of the interface. For example, a nearly equivalent solution with QuickTime video instead of Real video that also works with SVG for the slide representation and a Flash 6 based thumbnail overview component for faster slide loading and interface responding. This approach of mixing technology did not solve the problems

either. The reason was that the users had to install another plug-in, QuickTime instead of Real as well as the Flash plug-in. Moreover the reaction time of the interface could not be improved by this approach.

As a preliminary workaround, plug-in and browser checks had been added to the original version. These measures alert users if they try to use the interface with wrong software settings and thus reduced bug reports that are due to accessing the interface with wrong software setting. Also, a number of enhancements had been added to avoid unnecessary loading of slides when slide changes happen at a high frequency.

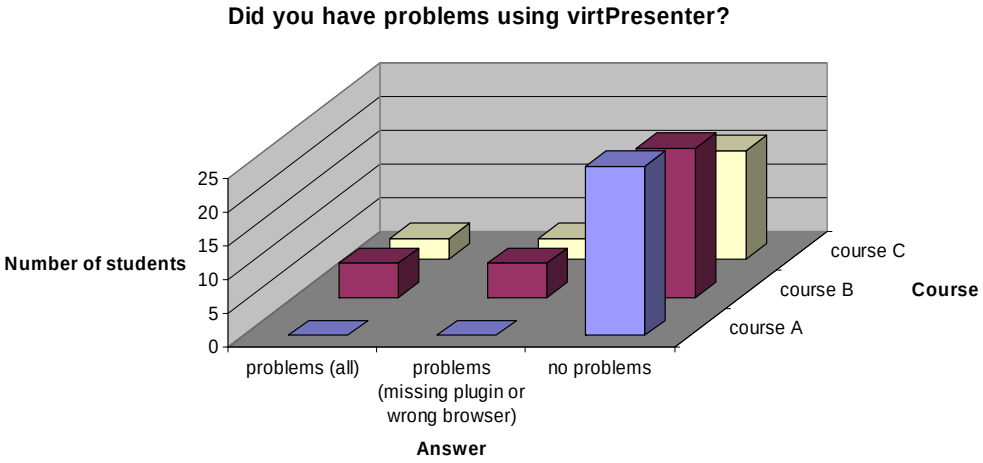


**Figure 4: User problems**

These approaches have, however, been limited by the technology setting in which they had been employed. In order to overcome these problems, we have turned to Adobe Flex 2 in combination with the Open Source Red5 streaming server backend as described in section 5.

## 4. TECHNOLOGY REVIEW: FLASH VS. SVG

In a strict sense, the new interface cannot reach the function range of the old virtPresenter interface described in (Mertens et al. 2006 a; Mertens 2007) by now.

This is mainly due to the fact that the new version does not yet feature an automatically generated thumbnail slide overview which is crucial to a number of functionalities implemented in the SVG based version (Mertens et al. 2006 a; Mertens et al. 2004). The thumbnail overview is used both to visualize the connection of navigation actions to the structure of a talk (Mertens et al. 2006 a) and to allow structure based navigation on the level of animations within a slide. The latter is realized by clickable slide elements that allow for direct navigation to the replay position when the corresponding slide element first appeared on screen during the recorded lecture (Mertens et al. 2004).

However, the reimplementation was necessary due to frequent user problems with unsupported computer platforms, wrong browsers or browser settings or missing plug-ins. The underlying shared infrastructure (Mertens et al. 2007) was enhanced to export, besides different podcast formats flash content (Flash video and Flash slides) (Ketterl et al. 2007 b, 2006 b). Adobe's Presenter (formerly named Breeze) is now also a part of the automatic lecture recording production chain. This software component enables a fast PowerPoint to Flash conversion that could be fully automated as well. This software component was selected in this new process due to the fact that it is reliable and now even affordable for a smaller university project. Today there are some open source or commercial PowerPoint to Flash export systems besides the Adobe product on the market. However, Adobe Presenter currently seems to be the only system that fits into our automated production chain. The other systems could not be integrated in the automatic production chain as they could not be started from other programmes. A Problem with Adobe Presenter is constituted by the fact that this component exports only Flash 6 slides in the current version. The communication between old Flash objects and new Flash 9 objects is not ideal at the moment. Difficult is for example the handling of different old Flash version based slides in a Flash 9 application. A prototype version which also features slide based navigation is depicted in figure 5 on the left hand side.

Nevertheless, the time for post processing (video and slide conversion and slide text analysis and building all the required software files for the interface could be reduced from previously about three hours down to only about one hour for a 1.5 hour lecture. Of particular importance is here, that the flash video conversion is much faster than our previous Real video conversion. Our initial recording format here is still MPEG-2 because of the fact that this video format is of good quality and can be converted into many different video/audio formats in the post processing process.

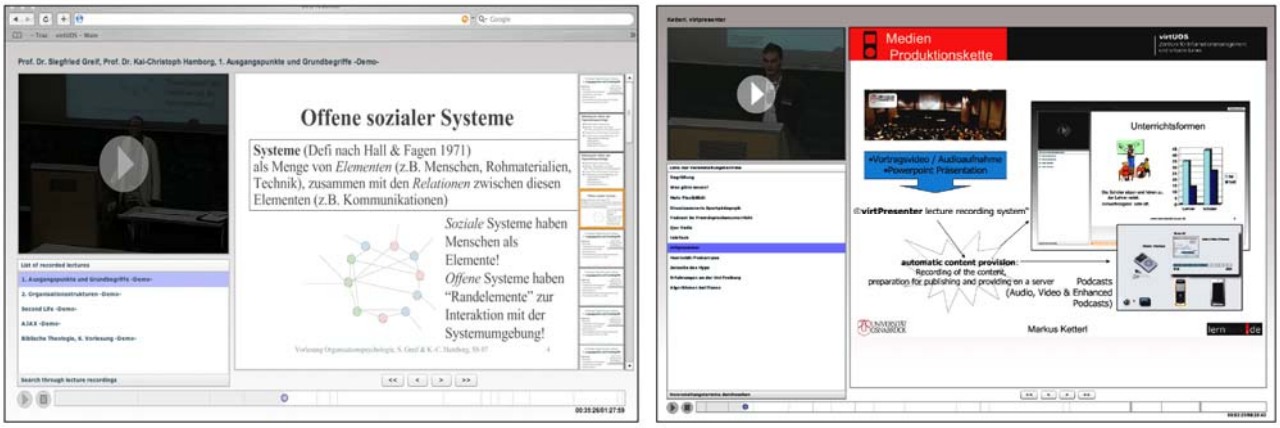


**Figure 5: virtPresenter 2 Flex technology based interface**

Figure 5 (right) depicts the revised and newly implemented Flex 2 based web interface. Besides the objective of using it on any computer platform without adjustments, the aim was that people without a technical background could use the interface as easily as internet experts. On the right hand side of Figure 5 one can find an area where users can choose from a list of recorded lectures or search text in the recordings. Figure 6 shows this lecture list (section a) and search results (section b) in a more detailed view. The lecture list gets updated over an RSS notify mechanism. Inspirational were our positive experiences with Apple's iTunes, their popular Music Store and the podcast subscriber facility (Ketterl et al. 2006 a, 2006 b). The main reason why we do not use Apple's iTunes (or other podcatcher software) and the podcast technology as main distribution facility is, that the navigation possibilities in podcasts are limited compared to the navigation options in the virtPresenter system. Further inquiries about navigation in lecture podcasts and how lecture podcasts are being used in contrast to the normal lecture recordings are ongoing. Several examination results with student users and external users are described in (Schulze et al. 2007) for virtPresenter and (Hürst and Welte 2007 a) for a system used at the University of Freiburg. In the revised virtPresenter system, users can subscribe to lecture recordings using our internal university learn management system Stud.IP (www.studip.de). The virtPresenter interface gets updated and shows the lecture recordings as soon as they are available. Aside from that, external users can subscribe to the recordings (like subscribing to a normal podcast with a podcatcher software like Apple's iTunes) and can view recordings for example that are open for public viewing. This lecture recording offer is presented over a public website. In short, this means that students as well as external viewers use the same interface for different recordings. They do not need to switch between applications and there is no need to follow additional links in other browser windows. The interface can also be used if a link from our lecture website or the LMS points to a specific lecture or a specific time index in a recording. This is done by interpreting assigned url parameters. The feature is a further extension of a functionality implemented for the SVG based version and described in further detail in (Mertens et al. 2005).

Section b in figure 6 also depicts a possibility to search in the recordings. Users can search not only in one web lecture but in all recordings they have subscribed to. The search results are presented in a hierarchical tree overview similar to Adobe's Acrobat. The results can be selected and are linked directly to the corresponding lecture recording section.

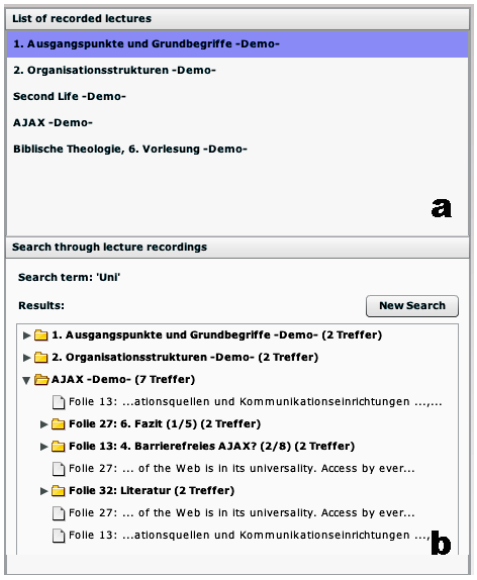


**Figure 6: RSS updated lecture overview with lecture search**

Due to the changeover to Flex 2 technology, users can navigate fluently in the recordings with a new time scrubber component (see figure 7). In the SVG based version, visible scrolling in the sense of (Hürst and Müller 1999) was only possible with the slides used in the recording, in the Flex 2 based version, it is possible for both slides and video. Presently we highlight slide borders in the timeline and show the lecture slide title directly above the respective area of the timeline. Colour-coded are the sections which have been viewed already by the user. When a lecturer is using the mouse cursor during the presentation, this data is also logged with the underlying recording system and the data can be presented in the user interface as well.


**Figure 7: Timeline with slide border visualization and slide title overview**

The Flex based interface responds considerably quicker than the old one (see table 1). Delays resulting from slide loading, jumps to other sections or disturbing video buffering that we had in the old Real video respectively SVG based version are not noticeable anymore. Even a complete reload of the system due to a browser refresh is quick. The interface was tested on Windows, Linux or Mac OS X computer platforms, all with the Flash 9 player plug-in. The results described were alike on all platforms.

## 5. VIRTPRESENTER 2.0: HOW FLASH 9 MADE THINGS FAST

For the new implementation of the lecture recording system we used Adobe's Flex 2 technology (this technology was introduced in June 06) for the user interface and for user interaction. Flex 2 is based entirely on ActionScript 3, which was introduced as a revised and extended programming language as part of Adobe's new Flash 9 player. Flex applications are deployed as compiled byte code that is executed within the Flash player runtime. The core of Flex is the developer-centric Flex framework, a library of ActionScript 3 objects that provide a foundation for building rich internet applications. Writing applications with Flex is similar to developing in .NET or Java (Kazoun and Lott 2007). Also, Flex provides a wealth of useful components so that developers do not have to build everything from scratch. Important besides the comfortable developer framework in our scenario is that neither a special browser version nor a combination of different plug-ins has to be installed on the users' computers (like needed in the SVG based implementation). The user only needs the Flash player plug-in for viewing the web lecture recordings. The current plug-in version is Flash 9, which is available for browsers on Windows (IE, Firefox and Opera), Apple (Safari, Firefox) and Linux (Firefox) as well. Normally this plug-in can be installed without difficulties or special computer knowledge. Besides, this software component is very popular and widespread nowadays (Téllez, A., G. 2007). That means that no special browser adjustments or compatibility checks are required. The same version will work on different computer platforms as a cross browser solution. The plug-in base for ActionScript 3 is a newly implemented virtual machine called AVM2 (ActionScript Virtual

Machine 2) that converts byte code into native machine code. It is more like a Java VM (Java Virtual Machine) or the .NET CLR (Common Language Runtime) than a browser script engine. The most important advantage is (and this is a main reason why we are using Flash 9) that the new browser environment is faster than previous versions and it uses much less memory on the computer (Adobe 2007). We could confirm this assertion in our daily work with the new Flex 2 framework. Student users report that they like how fast the new interface responds and reacts to user interaction. Further user acceptance/problem surveys are planned for February 2008.

In order to respond fast, a further component is important. Like mentioned before, a main problem was the video buffering of the Real player in the interface. A dedicated and reliable video server is also required. Like most universities we have a fairly good server infrastructure backend. Through that we could use Adobe's recommended and expensive Flash Media Server 2 for working with recorded lecture videos. Instead of this expensive solution we a have used an open source Flash streaming server implemented in Java for a couple of months now which is called Red5 (Red5 2007). The adoption was an experiment, because this open source server deployment was not really stress tested, barely documented and only available in version state 0.6 (currently version 0.6.3 is available). The server worked very stable even during the critical exam time at the end of the term.

Our productive streaming system used during that time was a 2,8Ghz Intel Dual Core Xenon processor based Windows XP system with 4 GByte RAM. This video server system is more than adequate with sufficient reserves in case of user request peaks. At present there is no need to use Adobe's expensive Flash Media Server 2 solution in our production environment.

## 6. BEHIND THE SCENES: ADMINISTRATION AND WORKFLOWS

Lecture recording with virtPresenter makes use of a fully automated recording and an extended production chain described in (Ketterl et al. 2007 b). While this process is fully automated, a number of administration tasks still remained. Currently we manage and generate eighteen web lecture recordings with additional podcasts (Ketterl et al. 2006 a) from different university courses in different rooms a week plus some additional recordings for special occasions like conferences and workshops with this system. This number increases steadily. The lecture recording system is tightly connected to the learn management system Stud.IP used at the University of Osnabrück. We have also defined more general interfaces that make metadata like the name of the course, the name of the lecturer and data for full text search available to other systems like content portals or search engines. These interfaces also allow for authentication handling by the other system. Thus users do not have to log in separately in the lecture viewer since they are authenticated externally, e.g. by the portal.

Normally the recordings are assigned to the web-page of the course in the university LMS. Figure 8 shows what this integration looks like in our university LMS Stud.IP.


**Figure 8: Lecture recordings in the Learn Management System**

The recordings can additionally be tagged with further meta-data or can be stored in other database systems wherefrom further platforms can use them as well. At present we are working on a rights management system for the recordings that will serve the purpose of defining whether episodes are available for university members,

publically (distribution over Apple's iTunes music store (Ketterl et al. 2006 a) for example), as part of a course exchange programme with other universities or on a pay per view basis.

A recurring administration task at the end of a study term is to bring the web lecture recordings offline on a computer DVD or a CD for data backup purposes, or for students respectively lecturers whishing to watch the lecture recordings offline. The normal approach in our production system was to copy the recorded video, the lecture slides and the complete source code for the web interface on that offline medium. In addition to the fact that it is not very convenient for users to start the recordings by clicking a specific file link in the DVD file system we had the drawback that the complete (maybe copyrighted) material is on that offline medium as well. Over the internet, we had at least user authentication to protect the content. A more attractive and promising way to reduce administration effort and to keep the content protected is to use Adobe's new integrated runtime environment called AIR (prior development name Apollo). AIR stands for Adobe Integrated Runtime.

The environment is a new cross-platform desktop runtime that allows web developers to use web technologies to build and deploy Rich Internet Applications and web applications to the desktop (Chambers et al. 2007).

During the last years, there has been an accelerating trend of applications moving from the desktop to the web browser. With the maturation of the Flash Player runtime and Ajax type functionality it became possible for developers to offer richer application experiences without disturbing page refreshes. This means that the Flex implementation of the web lecture system can be installed offline on a Windows PC or on a Macintosh system (a Linux version is promised by Adobe to appear by the end of 2007) and it will behave like any other application on the system. On Windows, for example, the virtPresenter web lectures appear now offline in the start menu and in the windows taskbar. As a drawback, users have to install the AIR runtime on their system.

The adoption of this technology in general is still in question. Why should users prefer a web like application on their normal desktop computers? Unlike this approach there are other projects and ideas that focus on the web as an operating system (Vahdat et al. 1996) or new alternative technologies as described in the next section.

In the literature one can find further examples for using RIAs on the desktop or ideas for adopting this technology (Chambers et al. 2007). In our lecture recording production environment, AIR solves some of the offline related problems. We can offer virtPresenter recorded AIR versions for standard download in case of a Red5 streaming server breakdown. Another prospect is that users do not need to be online while watching the lecture recordings since the AIR application could include all required files. The offline application gets updated through a new interpretation of the associated RSS files whenever the computer is online and new data (new lecture recordings) can be transferred and updated in the offline version.

For a simple lecture recording data backup mentioned in the beginning of this chapter, AIR is not an option, due to the fact that the content is encapsulated in the AIR application and it is problematic to disassemble it.

## 7. CONCLUSION AND FUTURE RESEARCH

During the last few years, Flash has evolved into an ideal content format for web lectures. Especially the fact that both slides and video can be replayed with one single browser plug-in makes web lecture interfaces built upon this technology easy to use for almost anyone. This paper has demonstrated the feasibility of a Flex 2 based user interface for web lectures and it has shown that this technology can be used to improve usability and ease the administrative workload.

With AIR it is even possible to protect content in offline versions of a web lecture. Given the fact that AIR and AIR- or Flash-like approaches (Silverlight (Cohen 2007), the JavaFX family, or Google Gears) are rumoured to be supported by a number of mobile devices in the near future, AIR could also open more perspectives for interactive presentation of web lectures on mobile devices. If AIR on mobile devices worked just like conventional AIR applications, it would be possible to produce learning content that can be used for normal websites and for m-learning modules at he same time, that is without expensive device adjustments. Our lecture podcasts (audio, video and enhanced podcast versions) (Ketterl et al. 2006 a, b) were a step forward to support mobile users with fine granulated lecture recordings.

In combination with additional mobile self assessments as developed for the system presented here (Ketterl et al. 2007 a) and other systems (Hürst et al. 2007 b) learning on the go becomes possible. The podcast technology has a drawback at present for mobile learners. Mobile users cannot give feedback to the lecturer for example due to

technical limitations of devices and of the podcast technology. With full AIR support on mobile devices, it is likely that these problems could be solved easily as one AIR application could run on different platforms (mobile, internet and desktop).

Another branch we are pursuing in the Flex based version of the interface is implementing social navigation functionalities that had previously been tested in the SVG based version of the interface (Mertens et al. 2006 d). Flex 2 does, however, open new perspectives for social navigation in lecture recordings. The reduced loading times allow for editing and rearranging content on the client side without having to change its server side representation. It is also easier to embed the player in other web sites. To prove this, some of our lecture recordings and the newly implemented Flex 2 based virtPresenter interface have been integrated as an application in the social community Facebook. An issue that does still remain to be solved is how navigation can be facilitated in re-arranged and re-structured content.

## REFERENCES


Adobe Systems Incorporated (2007) Flex2 technical overview, Technical whitepaper, Available at: http://www.adobe.com/products/flex/whitepapers/pdfs/flex2wp_technicaloverview.pdf (retrieved December 2007).

Bieber, M. (2002) Hypertext. *Encyclopaedia of Computer Science (4th Edition)*. Ralston, A., Reilly, E. & Hemmendinger, D. (Edts.) Nature Publishing Group. 2000. pp. 799-805.

Chambers, M., Dixon, R. and Swartz, J. (2007) Apollo for Adobe Flex Developers Pocket Guide, Adobe Developer Library, O´Reilly, March 2007.

Chambers, M., Dura, D. and Hoyt, K. (2007) Adobe Integrated Runtime (AIR) for JavaScript Developers, Adobe Developer Library, O´Reilly, July 2007.

Cohen, B. (2007) Silverlight Technical Articles. Silverlight Architecture Overview, Technical whitepaper, Microsoft Corporation, April 2007, Available at: http://msdn2.microsoft.com/en-us/library/bb428859.aspx (retrieved December 2007).

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, USA, 1999, pp. 333 – 342.

Hürst, W. and Welte, M. (2007 a) An evaluation of the mobile usage of e-lectures podcasts, *Proceedings of the Mobility Conference on Mobile Technology*, Applications and Systems, Singapore, September 2007

Hürst, W., Jung, S. and Welte, M. (2007 b) Effective Learn-Quiz Generation for Handheld Devices, *Proceedings of the 9th Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2007)*, Singapore, September 2007, to appear.

Hoppe, U., Klostermeier, F., Boll, S., Mertens, R. and Kleinefeld, N. (2007). *Wirtschaftlichkeit von Geschäftsmodellen für universitäre Lehrkooperationen - eine Fallstudie*. Zeitschrift für E-Learning 2 (2007), volume 3. pp. 29-40.

Kazoun, C., Lott, J., Programming Flex 2, Adobe Developer Library, O´Reilly, April 2007.

Ketterl, M., Heinrich, T., Mertens, R. and Morisse, K. (2007 a) Enhanced content utilisation: Combined re-use of multi-type e-learning content on mobile devices, *IEEE Multidisciplinary Engineering Education Magazine*, Volume 2, Issue No. 2, June 2007, pp. 61-64.

Ketterl, M, Mertens, R. and Morisse, K. (2006 a) Alternative content distribution channels for mobile devices, *Microlearning International Conference on Micromedia & eLearning 2.0: Getting the Big Picture*, Innsbruck, Österreich, 8.-9. June 2006. pp. 119-130.

Ketterl, M., Mertens, R., Morisse, K. and Vornberger, O. (2006 b) Studying with Mobile Devices: Workflow and Tools for Automatic Content Distribution, *World Conference on Educational Multimedia, Hypermedia &Telecommunications ED-Media 2006*, Orlando, FL, USA, Juni 2006, pp. 2082-2088.

Ketterl, M., Mertens, R. and Vornberger, O. (2007 b). Vector Graphics for Web Lectures: Comparing Adobe Flash 9 and SVG, Workshop on Multimedia Technologies for E-Learning (MTEL), *IEEE International Symposium on Multimedia 2007*, Taichung, Taiwan from December 10 to 12, 2007, pp. 389-395.

Lauer, T. and Ottmann, T. (2002) Means and Methods in Automatic Courseware Production: Experience and Technical Challenges, *In Proceedings of the World Conference on E-Learning in Corp., Govt., Health. & Higher Education*, E-Learn 2002, Montreal, Quebec, Canada, October 15-19 2002, pp. 553-560.

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 Corporate, Government, Healthcare & Higher Education, E-Learn 2006,* Honolulu, HI, USA, 13.-17. October 2006. pp. 2929-2936.

Mertens, R., Farzan, R. and Brusilovsky, P. (2006 d) Social Navigation in Web Lectures. *ACM Hypertext 2006,* Odense, Dänemark, 23.-25. August 2006. pp. 41-44.

Mertens, R., Friedland, G., and Krüger, M. (2006 b) To See or Not To See: Layout Constraints, the Split Attention Problem and their Implications for the Design of Web Lecture Interfaces, *World Conference on E-Learning, in Corporate, Government, Healthcare & Higher Education, E-Learn 2006,* Honolulu, HI, USA, 13.-17. October 2006, pp. 2937-2943

Mertens, R., Ickerott, I., Witte, Th. and Vornberger, O. (2005) Entwicklung einer virtuellen Lernumgebung für eine Großveranstaltung im Grundstudium, *Proceedings of the Workshop on elearning 2005*, HTWK Leipzig, 11.-12. Juli 2005, S. 197-210.

Mertens, R., Ketterl, M. and Vornberger O. (2006 c) Interactive Content Overviews for Lecture Recordings, *IEEE ISM 2006 Workshop on Multimedia Technologies for E-Learning (MTEL*), San Diego, California, USA, December 11to13, 2006, pp.933-937.

Mertens, R., Ketterl, M. and Vornberger, O, (2007 a) The virtPresenter lecture recording system: Automated production of web lectures with interactive content overviews, *In International Journal of Interactive Technology and Smart Education (ITSE),* 4 (1). Troubador publishing, UK, February 2007, pp. 55-66.

Mertens, R. Schneider, H., Müller, O. and Vornberger, O. (2004) Hypermedia Navigation Concepts for Lecture Recordings, *World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, E-Learn 2004*, Washington DC, USA, Nov. 2004, pp. 2480-2487.

Mertens, R. (2007) Hypermediale Navigation in Vorlesungsaufzeichnungen: Nutzung und automatische Produktion hypermedial navigierbarer Aufzeichnungen von Lehrveranstaltungen, PhD thesis, Universität Osnabrück, November 2007.

Red5 Open Source Streaming Server (2007), Available at: http://osflash.org/red5 (retrieved December 2007).

Schulze, L., Ketterl, M., Gruber, C. and Hamborg K.,C. (2007) Gibt es mobiles Lernen mit Podcasts? Wie Vorlesungsaufzeichnungen genutzt werden, *5.e-Learning Fachtagung Informatik (DeLFI)*, Siegen, Germany, September 2007, pp. 233-244.

Téllez, A., G. (2007) Authoring Multimedia Learning Material using Open Standards and Free Software, *IEEE International Symposium on Multimedia 2007 Workshop on Multimedia Technologies for E-Learning (MTEL),* Taichung, Taiwan from December 10 to 12, 2007, pp. 383-389.

Vahdat, A., Dahlin, M. and Anderson., T. (1996) Turning the Web Into a Computer, Technical report, University of California, Berkeley, 1996.

Ziewer, P. and Seidl, H. (2002) Transparent TeleTeaching , *19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE),* Auckland, New Zealand, Dec. 2002, (2), pp. 749–758.