Vector Graphics for Web Lectures: Comparing Adobe Flash 9 and SVG

Markus Ketterl
Virtual Teaching Support Center, University of Osnabrück, Schloßstrasse 9, 49069 Osnabrück
Email: markus.ketterl@uni-osnabrueck.de
Germany

Robert Mertens[1]
Fraunhofer IAIS, Schloß Birlinghoven, 53754 Sankt Augustin
Email: robert.mertens@iais.fraunhofer.de
Germany

Oliver Vornberger
Dept. of Computer Science, University of Osnabrück, Albrechtstrasse 28, 49069 Osnabrück
Email: oliver@uni-osnabrueck.de
Germany

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, not very common. A major reason for this is that multiple media formats have to be combined in order to replay video and slides. This paper 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.

1. Introduction

Vector based graphics formats offer a number of possibilities for the realization of web lecture interfaces for slide based talks. One mayor advantage is that they support capturing contents in a symbolic manner which is a requirement for searching in a recording [11]. 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 [12]. They also can be used to tackle a number of layout problems as described in [13].

Vector graphics are, however, not very common in web lectures. This paper 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 in productive use since March 07.

The article is organized as follows: Section 2 presents a brief analysis of experiences made with the SVG based interface version. Section 3 describes the new Adobe Flex technology used for the Flash based implementation. Section 4 introduces the Flex framework based interface and discusses the implementation details. Section 5 shows a possibility to use web lecture recordings offline. Section 6 shows future trends affecting the used technology and section 7 briefly summarizes the work presented in this article and refers to future projects and ideas.

[1] The author's work on this topic was conducted while he was employed at the Virtual Teaching Support Center, University Osnabrück, Schloßstrasse 9, 49069 Osnabrück

0-7695-3084-2/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.Workshops.2007.71

2. Experiences with the SVG-based Version

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 [16].

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. Things have, however, developed in a different direction.

While all the features described in [16] 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. As some interactivity and animation features of SVG had been used in the interface, replay was only possible with the Adobe SVG Viewer 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 Player buffers data when users navigate in the video. This buffering also slows down the interfaces reactivity noticeably.

Another problem with SVG was that the plug-in required only exists for the IE. 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 to user acceptance. In order to counter these effects a number of improvements had been devised for the SVG based version of the interface. These include compatibility checks for viewer plug-ins and a number of workarounds to speed up the interfaces reactivity. 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 technology as described in the following sections.

3. Adobe Flex Technology

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 user interaction. Flex 2 is written entirely in ActionScript 3, which was introduced as part of the Flash 9 Player plug-in. Flex applications are deployed as compiled byte code that is executed within the Flash Player runtime system.

The core of Flex is the developer-centric Flex framework, a library of ActionScript 3 objects that provide a great foundation for building rich internet applications. Writing applications with Flex is similar to developing in .NET or Java [7].

Also, Flex provides a wealth of useful components so that developers don't have to build everything from scratch. Important besides the comfortable developer framework is that neither a special browser version nor a combination of different plug-ins has to be installed on the users computers (like we needed in the SVG based implementation). The user only needs the small 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. That means that no special browser adjustments or compatibility checks are needed. The same version will work on different computer platforms.

The plug-in base for ActionScript 3 is a newly implemented virtual machine called AVM2 (ActionScript Virtual Machine 2) that turns byte code into native code. It is more like a Java VM or the .NET CLR than a browser script engine. The most important advantage is (and this is a main reason why we were using Flash 9 and not the deprecated Flash 8 technology) that the new browser environment is faster than the previous VM and it uses much less memory on the computer system [1].

We could confirm this assertion in our daily work with the new AVM2 and the Flex framework. Student users report that they really like how fast the new interface responds and reacts to user interaction.

In order to respond fast a further component is important. 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 the recorded lecture videos. Instead of this expensive solution we used an open source Java implemented Flash server during the past six months which is called Red5 [17].

The adoption was an experiment due to the fact that the server version wasn´t really tested, certainly barely documented and only available in version 0.6. The server worked stable even during the exam time in the end of the semester. At present there is no need to use the expensive Flash Media Server 2 in our production environment.

4. Implementation and re-work of the new Flex 2 based user interface

In a strict sense, the new interface cannot reach the function range of the old virtPresenter interface described in [12, 15] by now.

This is mainly due to the fact that the new version does not yet feature a thumbnail slide overview that is crucial to a number of functionalities implemented in the SVG based version [12, 16]. The thumbnail overview is used both to visualize the connection of navigation actions to the structure of a talk [12] and to allow structure based navigation on a 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 [16].

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 [14] was enhanced to export flash content (Flash video and Flash slides). Adobe's Presenter (formerly named Breeze) is now also a part of the lecture recording production chain. This software component enables a good and fast PowerPoint to Flash conversion that could be fully automated as well. Problematic is here that this component exports only Flash 8 slides in the current version. The communication between old Flash 8 objects and new Flash 9 objects isn´t ideal at the moment. Difficult is for example the handling of different Flash 8 slides in a Flash 9 based application.

Nevertheless our post processing time (video and slide conversion and slide text analysing) could be reduced from previously about three hours down to only one hour for a 1.5 hour lecture. Of particular importance, besides the slide transformation is here, that the flash video conversion is much faster than our previous Real video conversion. Our starting recording format here is still MPEG-2.

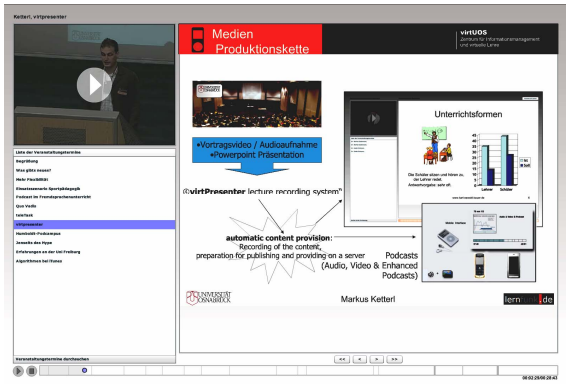

Figure 1: Reimplementation of the virtPresenter user interface component

Figure 1 depicts the revised and new implemented Flex framework 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 left hand side of Figure 1 one can find an area where users can choose from a list of recorded lectures or search text phrases in the recordings.

Figure 2 shows this lecture list (section a) and search results (section b) in a more detailed view.

The lecture list gets updated over a RSS notify mechanism. Inspirational were our positive experiences with Apple´s iTunes, their popular Music Store and the podcast subscriber facility [9, 10].

The main facts why we don't use Apple's iTunes 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 are ongoing. Several examination results with student users and extern users are described in [18].

In the revised virtPresenter system users can subscribe to lecture recordings using our internal LMS system studIP [www.studip.de]. The interface gets updated and shows the lecture recordings as soon as they are available. Aside from that, external users can view recordings that are open for public viewing. These recordings are presented over a public website. In short, this means that students utilize the same interface for different recordings. They don´t 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

special recording time. This is done by interpreting assigned url parameters. The feature is an extension of a functionality implemented for the SVG based version and described in further detail in [5].
Section b in figure 2 also depicts a possibility to search in the recordings. Users cannot search only in one web lecture but in all recordings they have subscribes to. The search results are presented in a tree overview like in Adobe's Acrobat. The results can be selected and are linked directly to the corresponding lecture section.

Figure 2: RSS updated lecture overview with lecture search

Due to the changeover to Flex 2 technology in conjunction with the Red5 streaming server, users can navigate fluently in the recordings with a new time scrubber component (see figure 3). In the SVG based version, visible scrolling in the sense of [6] 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. Color-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 3: Timeline with slide border visualization and slide title overview

The Flex based interface responds very fast. 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 aren't noticeable anymore. Even a complete reload of the system due to a browser refresh is quick. The interface was tested on Windows, Linux or MacOS computer platforms, all with the Flash 9 player plug-in. The results described were alike on all platforms.

5. Web lecture recordings offline

A recurrent administrator task at the end of a study term is bringing the web lecture recordings offline on a computer DVD or a CD for a data backup, 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 content and source code encapsulated 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 [3].

During the last years, there has been an accelerating trend of applications moving from the desktop to the web browsers. 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 [13].
This means that we can use the implemented Flex web lecture system and install it 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.

Questionable is the adoption of this technology in general at the moment: why should users prefer a web like application on their normal desktop computers? Unlike this approach there are other projects that focus on the web as an operating system [19] or new alternative technologies as described in the next chapter. In the literature, one can find further examples for using RIAs on the desktop or ideas for adopting this technology [3]. In our lecture recording production environment, AIR solves some of our offline related problems. We can offer virtPresenter recorded AIR versions for standard download in case of a Red5 server breakdown. Another prospect is that users don't 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 again and new data (new lecture recordings) can be transferred and updated in the offline version.
For our own lecture recording data backup mentioned in the beginning of this chapter, AIR is not an option at the moment, due to the fact that the content is encapsulated and it is problematic to disassemble it.

6. Future Trends

Apparently, a new trend is the fusion of the normal computer desktop and the internet. Both directions (web applications to the desktop and vice versa) are interesting at the moment. Adobe's integrated runtime AIR aims at web applications running on the desktop whereas Microsoft's WPF (Windows Presentation Foundation) for example focuses desktop applications operating in a web browser. As an alternative to the Flash player plug-in, Microsoft is going to promote a new Flash like browser plug-in called Silverlight [4]. Furthermore, Java is also going in a similar direction with its new JavaFX family. Also Google has shown examples of how web applications can be used without an internet backend. This Google browser plug-in architecture is called Google Gears.
Exciting are these new integrated runtimes and browser plug-ins particularly in conjunction with future mobile devices. If these runtime systems could be used on mobile devices, dependencies on available (maybe expensive) internet network connections could be reduced. Mobile applications could be updated or can communicate with other applications and servers as soon as a free hotspot is available. Silverlight technology has yet to be adopted exemplary to some mobile devices for example.

This development could also be interesting for watching lecture recordings on mobile devices. Our lecture podcasts (audio, video and enhanced podcast versions) [9, 10] were a step forward to support even mobile users with fine granulated lecture recordings or with mobile self assessments [8] in combination with lecture podcasts. The podcast technology has a drawback at present for mobile learners. Users cannot give feedback to the lecturer for example due to technical limitations of devices and of the podcast technology.

A next big step is that Adobe has announced Flex 3 as open source framework in June 2007.

7. Conclusion

The change-over and reimplementation of our lecture recording system was an important step towards increasing performance and thus further system acceptance. With Adobe's new Flex 2 (Action Script 3 based) framework in conjunction with technologies like the described AIR technology we are able to offer lecture recordings platform independently for different browsers in an online version. The recordings can be made available as both online and offline versions of the virtPresenter. Both can be updated through RSS notifications. With the adoption of the podcast technology during the last year we have been able to support even mobile users with lecture recordings [8, 9]. Disregarding smaller code changes, the newly implemented system is in productive use since March 2007. A survey in August this year is planned to show us, what the majority of the users think about the new technology.

8. References

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

[2] Chambers, M., Dixon, R., Swartz, J., *Apollo for Adobe Flex Developers Pocket Guide*, Adobe Developer Library, O´Reilly, March 2007.

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

[4] Cohen, B., "Silverlight Technical Articles. Silverlight Architecture Overview", Technical whitepaper, Microsoft Corporation, April 2007, Available at: http://msdn2.microsoft.com/en-us/library/bb428859.aspx

(retrieved August 2007).

[5] R. Mertens, I. Ickerott, Th. Witte, O. Vornberger „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.

[6] Hürst, W. & Müller, R., “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

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

[8] Ketterl, M., Heinrich, T., Mertens, R., Morisse, K., “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.

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

[10] Ketterl, M., Mertens, R., Morisse, K. & Vornberger, O. “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.

[11] Lauer, T. & Ottmann, T., “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.

[12] Mertens, R., Brusilovsky, P., Ishchenko, S. Vornberger O., “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.

[13] Mertens, R., Friedland, G., & Krüger, M., “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

[14] Mertens, R., Ketterl, M. and Vornberger O., “Interactive Content Overviews for Lecture Recordings”, IEEE ISM 2006 Workshop on Multimedia Technologies for E-Learning (MTEL), San Diego, California, USA, December 11-13 2006, pp.933-937.

[15] Mertens, R., Ketterl, M. Vornberger, O, “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.

[16] Mertens, R. Schneider, H., Müller, O., Vornberger, O., “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.

[17] Open Source Flash Server Red5, Available at: http://osflash.org/red5 (retrieved July 07).

[18] Schulze, L., Ketterl, M., Gruber, C., Hamborg K.,C., “Gibt es mobiles Lernen mit Podcasts? Wie Vorlesungsaufzeichnungen genutzt werden“, 5.e-Learning Fachtagung Informatik (DeLFI), Siegen, Germany, September 2007.

[19] Vahdat, A., Dahlin, M., Anderson., T., “Turning the Web Into a Computer”, Technical report, University of California, Berkeley, 1996.