# Lecture Recording: Structural and Symbolic Information vs. Flexibility of Presentation

**Daniel Stolzenberg[1] and Stefan Pforte[2]**
**[1]University of Rostock, Germany**
**[2]Baltic College Güstrow, Germany**
daniel.stolzenberg@uni-rostock.de
pforte@baltic-college.de

**Abstract:** Rapid eLearning is an ongoing trend which enables flexible and cost-effective creation of learning materials. Especially, lecture recording has turned out to be a lightweight method particularly suited for existing lectures and blended learning strategies. In order to not only sequentially playback but offer full fledged navigation, search and inspection of the recorded lecture, chapter marks and search indices have to be embedded. To solve this, two basic approaches for lecture recording tools can be identified - both of them having certain advantages and drawbacks. On the one hand there are systems based on symbolic representation of common slideshow formats like MS PowerPoint. Therefore, they preserve structure and symbol information contained therein, but are lacking flexibility of supported dynamic and interactive formats. On the other hand there are systems based on pixel representation and screen grabbing technologies. While supporting any presentation content, structural and symbolic information cannot be extracted directly and thus has to be post-processed from the recorded video. This paper discusses a perspective of combining these approaches by widening the slide-metaphor to a more flexible scene-based presentation, preserving both the structural and symbolic information. One possible attempt for this is identified by introducing a browser-based scene concept. Symbolic information can be directly extracted from the XHTML source code and structural information derives from switching through scenes. The browser itself is capable of presenting a wide range of dynamic and interactive formats, thus offering more flexible presentations. For approving the proposed concepts, a prototype called “Virtual Overhead” was developed and evaluated.

**Keywords:** rapid e-learning, lecture recording, lightweight content production, browser interactivity

## 1. Introduction

With the increasing availability of computers and growing bandwidth of internet connections technology-enhanced teaching and learning scenarios have shifted from research projects to wide spread usage. This results in the need of easy and cost-effective approaches and tools for content creation. Traditional development processes for eLearning content are quite complex and cost-intensive – ranging from 50.000 to 100.000 EUR per lecture hour (Lauer and Ottmann 2002) – and teams with expertise from different domains are required. *Rapid eLearning (REL)* strategies try to reduce this effort. Basically, these approaches have in common, that the *subject matter expert* (SME) is solely responsible for content-creation. In this he/she is assisted by special authoring tools, which include templates for both instructional and screen design. By abstracting from programmatic aspects no additional technical knowledge is needed. This approach reduces development cycles and costs, sceptics argue it also reduces the quality of the final learning content.

Lecture Recording is one particular REL-concept for cost-effectively creating learning content. By recording and transmitting face-to-face lectures or presentations, synchronous online attendance for distance learning is enabled and moreover lectures are preserved for later asynchronous playback. Especially suited for blended learning strategies, Lecture Recording is a way to elegantly combine traditional and technology-based learning scenarios. In addition it provides versatile digital learning materials at negligible additional costs and therefore can be described as *lightweight content production* (Kandzia et al. 2004).

In order to acquire every relevant aspect of the live presentation, three basic streams have to be captured. First and foremost, the teacher’s *verbal narration* which accompanies the *slides* or *visual presentation*. These two sources are considered being essential streams, whereas the *live video* should neither be overestimated nor underestimated in its importance for learning (Lauer and Ottmann 2002). The presenter’s video gives an impression of his presence and serves for improving authenticity, but conveys comparatively little information in most cases (Schütz 2003). The more expressive the lecturer’s gesture is, the more valuable the video gets for learners. In summary a modern system for lecture recording should integrate capabilities for dealing with video, as its importance will increase for bridging perceived gaps in distance learning.

In order to emphasize certain aspects of the visual presentation, a lecture recording system should integrate options to superimpose *annotations* like freehand drawing or rectangles over the content. In higher education and universities typically an electronic lecture of about 90 minutes is produced. Hence providing only

 
Reference this paper as:
Stolzenberg, D. and Pforte, S. “Lecture Recording: Structural and Symbolic Information vs. Flexibility of Presentation.” *The Electronic Journal of e-Learning* Volume 5 Issue 3, pp. 219-226, available online at www.ejel.org

sequential playback does not serve the learners needs, especially when used as additional material for preparing tests and exams. Instead users want to easily navigate within the electronic lecture (Zupancic and Horz 2002). Therefore common controls like *play* and *pause* should be completed with *random access* via *timeline navigation* controlled by a visual slider. Furthermore specialized interface controls make sense for replaying the lecture at a different speed while maintaining pitch. Hürst presents such a novel way of audio browsing with an elastic slider controlling the speed of replay (Hürst et al. 2005). Moreover learners need to directly locate and access certain points, like chapters or individual slides. For this *timestamps* or *indices* are used, which have to be embedded in the lectures media files and user interface. Additional *posterframes* corresponding to the timestamps can further simplify visual selection of relevant portions. In order to enable users to inspect the lectures content rapidly, search and retrieval features are required. Prerequisite for both navigation and search is the ability of the recording system to access and gather relevant symbolic and structural metadata from the presentation's content. *This requirement and its relation to the flexibility of presentation offered by the system is the key aspect of this paper and is further investigated in the remaining parts.*

## 2. Lecture recording approaches

To facilitate the identified features of navigation and retrieval, the process of capturing the presentation's content is of critical importance. While the recording of audio and video stays the same for different approaches, concepts of existing systems vary a lot regarding the visual presentation. Early suggestions were based on simply recording a blackboard or overhead projection with video cameras, which inevitably lead to unacceptable quality of the presentation (Lauer and Ottmann 2002). In contrast, modern lecture recording systems are part of an all-digital-environment and therefore can be grouped into different categories, regarding the *representation of content* (Ziewer 2006):

1. Symbolic Recorders (input grabbing)
2. Screen Recorders (output grabbing)
3. Hybrid Systems

### 2.1 Symbolic recorders

*Symbolic Recorders* (figure 1) make use of a priori knowledge concerning the presentation's *input* to obtain needed metadata for navigational indices and search functionality. These systems directly store content in symbolic representation, which has to be interpreted during any replay. Hence, a special player software has to be used, which makes it possible to adapt playback to different devices. Textual symbol information of every slide is included in the final lecture media and enables searching. Events like switching through the slides of a presentation trigger the recorder to set timestamps for playback and slide-based navigation.

According to this, symbolic recorders must at least have a connection to the presentation software in order to receive this information. In most cases they are tightly integrated with the presentation software, thus vitally depending on it. Particularly, the flexibility of the presentation's content is determined by the presentation software.

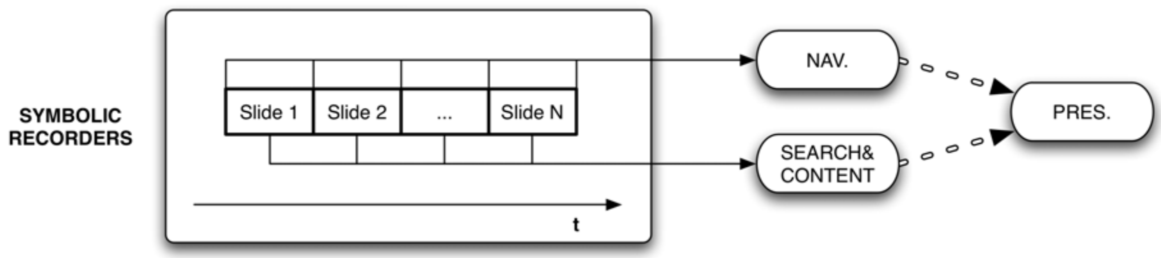


**Figure 1:** Symbolic recorders

### 2.2 Screen recorders

In contrast *Screen Recorders* (figure 2) are fully independent of the presentation content and software. They get the visual *output* by either directly grabbing the computers graphics buffer or its output via a VGA-grabber card. In return they do not have any prior information to directly derive required navigational and retrieval metadata, which has to be extensively post-processed from the recorded video stream. This independence in turn results in virtually unlimited flexibility of presentation. Besides conventional presentation software, every other program can be executed and recorded, which is especially useful for simulations, programming exercises and other sophisticated applications.

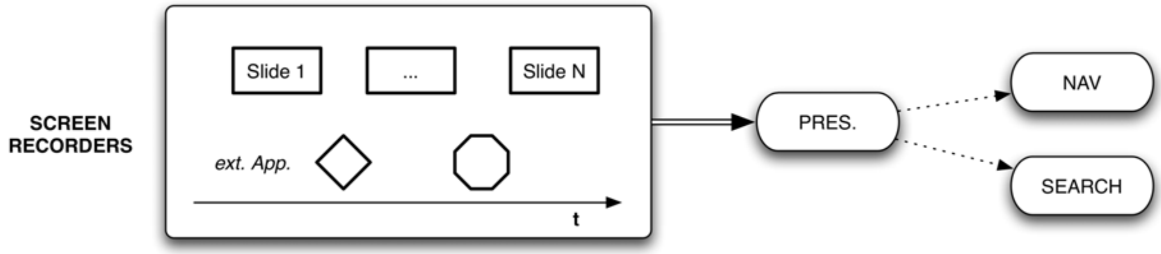


**Figure 2:** Screen recorders

## 2.3 Hybrid systems

*Hybrid systems* (figure 3) combine key aspects of these two approaches. While grabbing the visual presentation's output (screen recording), connecting this recorder with the presentation software directly provides symbolic information for searching and timestamps for navigation (symbolic recording). This concept permits mixed usage of the designated presentation software together with any other software. Of course derivation of navigation and search metadata is limited rather strictly and basically only covers the presentations software's content. Equal to pure symbolic recorders the presentation software of hybrid systems determines which input formats and features are supported to offer full derivation of needed metadata.

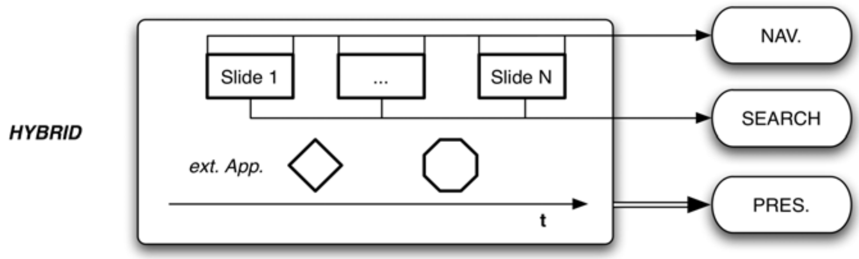


**Figure 3:** Hybrid systems

## 2.4 Comparison

A comparison of the concepts described above regarding metadata and flexibility of presentation shows that *screen recorders* offer the most flexible features for the lecturer but at the cost of metadata. While this information can be retrieved from the screen video by automated character/text recognition and slide detection (Ziewer 2004), such solutions are quite complex and results are not as reliable as directly extracted metadata. In exchange indices are collected for virtually any content including images, videos and animations. In contrast *symbolic recorders* offer high quality metadata at the cost of flexibility of presentation. Only features incorporated in the presentation software itself are supported. In summary, *hybrid systems* trade off drawbacks and advantages of both concepts by offering directly extracted metadata and usage of external software as well. Therefore this approach is especially suited for further investigation.

## 2.5 Integrating and controlling external content

Flexibility of presentation is the most important aspect for the SME when designing course content. But when it comes to actually giving the lecture, this flexibility has to be accompanied by means of structuring, arranging and coordinating various content sources. Common presentation programs like MS PowerPoint serve this requirement well by dividing the lecture into small segments called slides. They contain diverse media elements and can *easily be presented successively*. But unfortunately this kind of software has quite restricted support of input formats and aims at little interactivity during presentation (section 3).

In the previous section the concept of a hybrid system was selected, because it permits the lecturer to step out of the presentation software and use arbitrary software to augment the lecture with additional elements. But in doing so, the instructor is *solely responsible* for preparing every content source before the lecture and *managing the lecture's progress by selecting and advancing to the appropriate external content*. As this has to take place in real-time and parallel to verbally explaining the lecture's subject, certain pitfalls are inherent

 ISSN 1479-4403

to this approach. For example, a lecturer using web content might get “lost in cyberspace”, because the URL must be typed in or a link has to be placed somewhere. In addition, switching through applications causes visual discontinuities which in turn provoke irritations for learners, not knowing what to concentrate on. When several external applications are used, the instructor may have to switch through them in search for the intended one. Effects like these are challenging for both the instructor and the learners as soon as external sources are used more than occasionally.

In summary, neither existing presentation software provides enough flexibility, nor are external sources and software easy to control and integrate into a conventional slideshow presentation. To overcome this, it is desirable to give the lecturer options to control the presentation’s flow in a concise and consistent way. To achieve this, integrating extended features into the presentation software seems to be necessary, most notably interactive features. In addition, this approach renders even more sources accessible for direct metadata-collection in a hybrid lecture recording environment (figure 3), like extracting from XHTML-source code.

## 3. Browser-based presentation software

Modern slideshow presentation software in fact supports a vast number of media formats for audio, image and video assets. In addition a lot of assets can be created within the software itself, ranging from simple objects (lines, rectangles…) to complex flow charts. Even some limited animations can be set for visual elements and slides. But the overall design of slideshow software is originally focussed on static content for text- and image-centred (business) presentations. It therefore does not stretch out to comprehensive interactivity. For example, it is not intended to directly display PDF-content (like scientific papers) or to control parameters of interactive graphics. Moreover, web resources and services cannot be included directly, instead the presenter has to make screenshots and manually embed them into the slideshow. Or a dedicated browser has to be used for this, leading to problems of managing and controlling the lecture’s flow between the presentation itself and multiple external sources (as described in section 2.5). In addition desired metadata for navigation and retrieval cannot be extracted when using external browser software in a hybrid lecture recording system (section 2.3).

However, a modern web browser is capable of loading and displaying content of numerous formats and standards, either directly or via plug-ins. The support for audio, image and video files is comparable to conventional presentation applications. As common slideshow presentations can be exported to popular image formats, existing content can be imported and reused, although dynamic content gets lost. Indeed, experience shows that few presentations actually use these features beyond fading-in bullet lists step by step or slide transitions.

Due to browser plug-ins even future file formats and standards can easily be displayed without modifying the presentation application itself. The overall extensible architecture of browser engines allows sophisticated content like Java-applets to incorporate specialised elements suited for teaching purposes. For instance in mathematics or physics education interactive graphs of functions are often used and can be modified by controlling certain parameters. In contrast to slideshow applications browser-based software is ideally suited for this kind of purposes. Together with the trend of *rich internet applications (RIA)* and the upcoming *“Web2.0”,* the browser’s role and importance is expected to increase for almost every aspect of technology enhanced work and learning in particular. In conjunction with the hybrid lecture recording approach, it is sufficient to display the content, as the screen grabbing process merges several formats into the selected format for publishing. Thus, learners only have to support this final format and are independent of the presentation’s original input. Nevertheless it is desirable to give students access to the majority of learning materials for self-paced learning (especially interactive content), which can be accomplished by web-publishing in widespread formats. In return, these materials can be directly integrated into browser-based presentation software.

Together with these benefits towards flexibility some disadvantages result from using a browser-engine. At first there are known incompatibilities of browsers concerning display of standards (like Cascading Style Sheets) and every rendering engine handles content a little differently. Moreover images, which are intended to display full-screen, have to fit exactly into the browser’s view-port. Thus, they have to be created with these dimensions, for instance when exporting PowerPoint slides. Compared to these drawbacks the advantages seem to be more important.

 

## 4. "Virtual overhead"

By developing a prototype the suitability of a browser as a presentation software engine should be proved. But instead of just exchanging common display engines for a browser, further flexibility enhancements were aimed at. Therefore, a different guiding metaphor to "slide" was identified and shaped to match the requirements of instructors and lecturers.

### 4.1 Enhanced scene metaphor

Besides integrating suitable features and web-support into a conventional presentation program, a browser engine can straightforwardly serve as the basis of alternative presentation software. Although this approach cannot replace every external application, it can substantially expand the flexibility of the presentation software. To offer more interactive flexibility, basically the slide-metaphor of existing application can be widened to a *scene-metaphor.* A scene is the adoption of terms originated in theatre and film production. It consists of a stage populated with actors playing certain roles coordinated by the director. Scenes differ in background/setting and actors and intentionally have a dynamic and active character. Because of these attributes the term "scene" seems to be better suited for describing properties of the intended interactive and dynamic presentation segments.

Applied to browser-based presentation software, a scene is like a browser "tab" or window – a viewing port for any supported content obtained from a URL. If static formats are loaded it is equal to an ordinary slide, but in case of interactive content it can be freely manipulated, exposing rich features for teaching purposes. When preparing a presentation, scenes can be added and directed to a specific URL, which not necessarily has to be on a remote server but as well can be a local file. During the presentation, scenes are progressed step by step equal to slides.

Beyond using a browser-engine as basis, a scene concept suggests additional modifications of presentation application design. In most teaching situations questions of students lead to further explanations demanding space for additional freehand-drawings or references to supplemental materials. For instance, the instructor queries "Wikipedia" or "Google" concerning a topic arising from a student's request. As this could not be anticipated during preparation of the presentation, a new scene has to be created at lecture time. This flexibility strengthens the usage of a scene-metaphor even more – analogous to improvisation in theatre performances. This concept is followed throughout the overall design. From this point of view scene-based software should be *modeless*, so there is no difference between authoring and presentation (in contrast to common slideshow applications). This allows to give the presenter full control over the presentation's relevant parameters during the lecture.

The design of the annotation feature is closely connected to this modeless approach: It is modelled as a layered stack of elements on top of the scene content. In fact the annotations mirror the basic elements for static slide content, including freehand pen drawing, formatted text boxes, geometric primitives (rectangles, circles) and image support – all annotations provide alpha channels for transparency. Every element can be dragged independently enabling recomposition of scenes even during presentation, if the need for further variation arises. If the scene's content is scrollable, the annotations have to stay aligned with it. (Imagine a lengthy explanation by "Wikipedia", where only a paragraph of the last part is relevant to the lecture's topic and therefore was previously highlighted with a coloured box.) Moreover, the lecturer can easily drag & drop elements onto a new scene from a library of previously created/imported annotations. This concept allows slides to emerge visually and dynamically at presentation time instead of just switching, which is capable of focussing attention of learners to the process. This technique can be compared to a magnetic whiteboard with sets of adhesive icons and is often used in analog teaching scenarios. Because annotations are part of the scene, they disappear when scenes are switched, are made visible again when returning to that scene later during presentation and are even saved together with the presentation.

In conclusion flexibility of presentation is augmented in three aspects by applying a scene concept. At first, a browser-based scene can contain far more interactive content (JavaScript, Flash, Java-Applets, RIA …). Second, a modeless design offers full control to the presenter, even to add new scenes while giving the lecture. Third, the visual elements of the annotation layer completely mirror static slide elements and are freely drag-able during presentation.

### 4.2 Application design

In order to validate the concept of browser- and scene-based presentation software, a prototype called "Virtual Overhead" was developed. Furthermore, the goal was to combine this presentation software with a screen recorder according to the *hybrid lecture recording* approach identified in section 2. To reduce development time and to access special features, the prototype was implemented on Apple Mac OS X, which offers an easy-to-use and yet flexible programming environment, including a browser engine (based on Konqueror's KHTML engine) and far-reaching OpenGL support for screen grabbing. A straightforward Model-View-Controller (MVC) pattern was used to implement the system (figure 4). A "project controller" mediates between the primary model class ("project") and related functionality. For instance, it adds or removes scenes and responds to switching scenes. A "project" itself contains an array of "scenes" together with additional information (imported/created assets, bookmarks etc.). In turn, the "scene" is basically composed of a browser view-port associated with an URL and a stack of "annotation" items. All annotation elements (freehand, rectangle, etc.) are subclasses of a virtual "annotation" class. The internal state of a "scene" is managed by a "scene controller", it adds or removes annotations and redirects the URL associated with a scene. Finally a "tool controller" coordinates the function of the pointing device, whether the user is actually drawing, moving annotations or interacting with the browser's content. Moreover, it is responsible for maintaining and changing the visual properties (colour, thickness etc.) of created or selected annotation elements. The visual composition of the browser view-ports and the annotation layer is directly handled by the operating system with an independent transparent child-window.

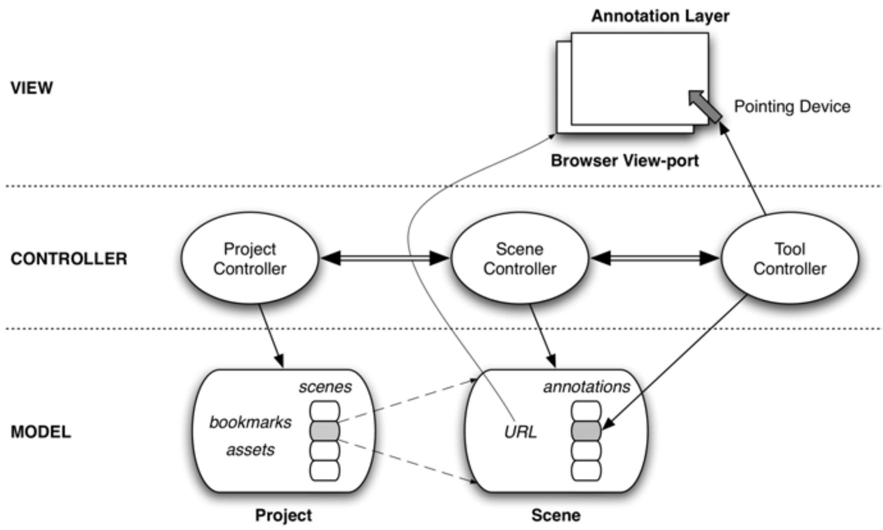


**Figure 4:** MVC Application Design

The screen recorder is implemented as a stand-alone application bundled with the presentation software. It is controlled via *distributed objects* – a system for inter-application-communication on Mac OS X. When a connection is established after launching from the main application, the recorder responds to requests like starting, pausing or triggering chapter marks (navigational indices). The grabbing process itself operates via a read-only full-screen OpenGl-context in conjunction with the *glReadPixel*-Function, which copies pixels from the video buffer.

### 4.3 Evaluation of criteria for lecture recording

To compare the developed system with common requirements and features of other systems, a catalogue of important criteria for lecture recording systems can be used (table 1). Peter Ziewer (Ziewer 2006) extracted and merged various aspects of existing catalogues (Lauer and Ottmann 2002, Mertens and Rolf 2003) to a final catalogue of eleven items. It does not claim to be complete and suitable for all kinds of evaluation, but reveals the most important aspects. The importance of certain criteria can be very different for several usage scenarios – for instance navigational indices are not important for synchronous electronic lectures in distance learning (Ziewer 2006).

**Table 1:** Criteria for Lecture Recording Systems (Ziewer 2006)

| | Criterion | Subcriteria | |
|---|---|---|---|
| C01 | Verbal Narration | | YES |
| C02 | Live Video | | YES |
| C03 | Presentation Content | a) Format of Presentation Documents | Common Formats + Plug-Ins + Web Standards |
| | | b) Supported Presentation Software | Any (but no metadata) |
| | | c) Supported Additional Applications | YES (Mac OS X only) |
| C04 | Annotations | a) Type of Annotations | Freehand, Rect., Circle, Icons, Text |
| | | b) Dynamic Capture and Replay of Annotations | YES |
| | | c) Annotation Associated with Segments | YES |
| | | d) Student Note-Taking | NO |
| C05 | Metadata | | indirect |
| C06 | Post-processing | a) Video-like Editing | YES |
| | | b) Content Editing | NO |
| | | c) Creation of Distributable Media | YES |
| C07 | Navigation | a) Structured Electronic Lectures | YES (per scene) |
| | | b) Random Access | YES |
| | | c) Visible Scrolling | YES (QuickTime) |
| C08 | Information Retrieval | a) Searchable Content | Planned |
| | | b) Range of Searchability | Planned |
| C09 | Format of Produced Electronic Lectures | a) Lossless Reproduction | NO |
| | | b) Scalability | NO |
| | | c) Streamability | YES |
| | | d) Format | QuickTime (+ Transcode) |
| | | e) File Size and Bandwidth | Several options |
| C10 | Platform Independency | | NO (Mac OS X only) |
| C11 | Synchronous Electronic Lectures | | Planned |

As already mentioned in section 1, the primary streams are the instructor's verbal narration (C01) and the visual presentation content (C03), whereas the live video (C02) is not that important, but should be supported. A key aspect of this paper is the flexibility of presentation, which is covered in this catalogue by the subcriteria C03 a) – c). *Especially in criterion a), the developed prototype differs a lot from existing systems due to its browser-based scene concept. It not only supports common file formats, but features technologies like Java and other dynamic web content.* Even an external presentation software can be used like any other application (C03 b+c), but in this case no metadata can be extracted directly. *The design of the virtual overhead's annotation feature (C04) is another aspect of distinction in comparison to existing systems. It mirrors common slide elements (C04 a+b) in a very dynamic and flexible way enabling the instructor to express in a wider range. Moreover annotations are bound to scenes (C04 c) and even are aligned with scrollable content.* Admittedly students' notes can not be included into the final lecture video. Metadata (C05) for the lecture itself (like Learning Objects Metadata – LOM) are supported only indirectly through video formats or html-pages used for distribution. Post-processing and distribution can be applied as with every video asset (C06 a+c), but due to the hybrid lecture recording approach (pixel-based), the content itself can not be edited (C06 b).

Another key aspect of this paper are navigation and retrieval features, which are covered by criteria C07 and C08. The hybrid lecture recording approach makes it possible to set navigational indices when scenes are switched during the presentation (C07 a), whereas random access and visible scrolling (C07 b+c) are supported through the video formats used for distribution (Apple QuickTime recommended). Although information retrieval and searching (C08) is not yet implemented, the system's design and the browser-based scene concept allow for comprehensive metadata, like extracting XHTML source code for search indices. As already mentioned, the lecture is distributed via several wide-spread pixel-video formats, which in most cases do not support lossless reproduction or spatial scalability (C09 a+b). The prototype presented in this paper was developed for Apple Mac OS X only in order to reduce development time and access special features, thus is not independent of platform decision (C10). But the overall concept and design can be transferred to a cross-platform base (like Mozilla). Finally, synchronous electronic lectures are planned, but not yet implemented.

## 5. Conclusion and future work

This paper started with a comparison of basic lecture recording approaches, which lead to the concept of a hybrid system as the most promising starting ground for further development. It allows navigational and search metadata to be gathered directly from the connected presentation software, but even enables flexible presentations by supporting external applications (section 2.4). Because additional programs cannot provide metadata directly and are difficult to integrate seamlessly into the presentation, this paper set up the following thesis: The more flexibility is directly incorporated into the presentation application itself, the easier the presentation is to control for lecturers and the more metadata can be extracted directly (section 2.5).

To accomplish this, a browser engine was proposed as basis for more flexible presentation software. It supports numerous wide-spread interactive formats and standards and is extensible in various ways. Moreover, it offers the possibility to immediately use web content. Together with this a shift of metaphors from “slides” to “scenes” was proposed to reflect the orientation on interactivity. In turn a “scene” metaphor suggested additional modifications like a modeless design (no distinction between authoring and presentation) and a highly dynamic annotation layer. In order to validate the concept of a browser-based scene concept, a prototype called “Virtual Overhead” was developed and will be used for the conference presentation. The prototype was compared to existing systems and common requirements. For this comparison a catalogue of criteria developed by Ziewer (2006) was used.

As not every desirable feature has been implemented yet, search functionality and retrieval have to be developed in near future, starting with extracting metadata from both XHTML-code and textual annotations. A suitable container for this data has to be identified together with the implementation of methods for user-driven search-queries. Furthermore, synchronous distance lectures have to be supported. The frames already grabbed for recording can as well be used for streaming, when properly compressed.

In summary, the proposed browser-based scene concept was identified as a possibility to offer more interactivity and visual expression to lecturers. Finally, this approach seems to be suitable to solve the contradiction between flexibility of presentation and structural or symbolic information.

## References

Hürst, W., Lauer, T., Bürfent, C. and Götz,G. (2005) Forward and Backward Speech Skimming with the Elastic Audio Slider, Proc. of the 19th British HCI Group Annual Conference (HCI 2005), Edinburgh, Scotland, UK.

Kandzia, P.-T., Kraus, G. and Ottmann, T. (2004) Der Universitäre Lehrverbund Informatik – eine Bilanz, GI Softwaretechnik-Trends, Vol 24, No.1, pp54-61.

Lauer, T. and Ottmann, T. (2002) Means and Methods in Automatic Courseware Production: Experience and Technical Challenges, Proc. of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-Learn 2002), No. 1.,AACE Press, Montreal, Canada, pp553-560.

Mertens, R. and Rolf, R. (2003) Automation Techniques for Broadcasting and Recording Lectures and Seminars, Proc. of SINN03 – Third International Technical Workshop and Conference.

Schütz, F. (2003) Producing eLearning materials on the fly – only a great dream? Proc. of the Second International Conference on Multimedia and ICTs in Education, Bada joz, Spain.

Ziewer, P. (2004) Navigational Indices and Full-Text Search by Automated Analyses of Screen Recorded Data, Proc. of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-Learn 2004), AACE Press, Washington, D.C., pp3055-3062.

Ziewer, P. (2006) Flexible and Automated Production of Full-Fledged Electronic Lectures, PhD-Thesis, TU München

Zupancic, B. and Horz, H. (2002) Lecture recording an its use in a traditional university course, Proc. of the 7th Annual Conference on Innovation and Technology in Computer Science Education, ACM Press, New York, USA, pp24-28.

 