Anchored Discussions of Multimedia Lecture Recordings

Tobias Lauer, Stephan Trahasch and Bernd Zupancic
Institute of Computer Science, University of Freiburg, Georges-Koehler-Allee, 79110 Freiburg, Germany
{lauer, trahasch, zupancic}@informatik.uni-freiburg.de

Abstract – Lecture recording has become a widespread way of producing learning contents rapidly. However, the resulting documents have some disadvantages regarding learner interaction, in particular regarding collaborative learning based on such documents. This paper proposes a model for anchoring group discussions in learning contents in a fine-granular way. Artifact-centered discussion is regarded as collaborative annotation of documents, with an extended data model for annotations which is independent of document and media types. After outlining the generic framework and data model, we describe its application to the scenario of group discussions around lecture recordings. Notes can be anchored both spatially and temporally in the multimedia documents. The infrastructure for sharing notes has been implemented as an Annotation Web Service (AWS), and the player software for viewing recordings has been extended into a client providing thread-based as well as document-centered discussion features.

Index Terms – Lecture Recording,Annotations, Anchored Discussion.

INTRODUCTION

Presentation recording has gained increasing interest both as a method for preserving live lectures and as a means for rapid content production [8]. Live lectures, presentations, and training units given in front of an audience are captured and turned into multimedia documents that can be used for e-learning. The idea of automatically recording computer-based live presentations in order to produce multimedia learning contents has been studied for nearly a decade, and a wealth of software systems has been developed to support it [6]. The main goal is to capture the relevant information streams created in a live presentation (slides, audio, video, notes, etc.) and deliver them to learners as an integrated multimedia document with rich structural and navigational functions.

Among the factors making the approach attractive are its cost-efficiency and the very quick availability of the produced contents. Instructors simply prepare the materials (such as slides, animations etc.) as usual and teach their regular face-to-face lectures, which are recorded automatically by the hardware and software in the lecture hall. In the ideal case, no additional personnel are required and the instructors do not even notice that they are being recorded. Depending on the time required for post-processing and format conversion, the resulting multimedia documents can be ready within minutes after the live class has ended.

Learners can access these e-lectures via download or streaming over networks or access them from mass storage media like CD-ROM/DVD [5]. Lecture recordings can be used as content for distance learning courses or as supplement for on-campus lectures (mixed mode). Over the last couple of years, we have obtained considerable experience with offering distance courses over the Internet based on lecture recordings and in the mixed mode through projects like [13][14]. Even those students attending the live lectures make extensive use of lecture recordings when recalling the content and preparing for examinations [15].

A problem found in the context of lecture recordings is that learner interaction has some limitations. First, interacting with the contents is restricted to certain forms of navigation and search. Second, user interaction is completely optional, as opposed to many computer-based courses, where quizzes etc. force learners to interact frequently. And third, users can only interact with the computer, not with other co-learners. The reasons for these limitations can partly be found in the nature of the documents themselves, which – by way of their production method – are rather linear and expository. Even though they do not have to be watched in a linear fashion, there is only limited room for active exploration. Other reasons are not restricted to lecture recordings but also apply to other computer-based courses. For example, it is usually not possible to annotate dynamic, time-dependent multimedia presentations similar to the way printed text and graphics can be annotated. In addition, opportunities for cooperative learning with such documents in real-life scenarios are rather sparse. If collaborative features are provided, they are usually separated from the contents. For example, most learning management systems offer chat tools or discussion forums for user interaction and collaboration. However, they are not directly accessible from or linked to the learning contents and learners have to switch back and forth between the two. Moreover, communication tools are often provided without any instructions or guidance, and learners do not know what they are expected to do with them.

This paper describes a model for anchoring discussion in the learning contents. While the presented framework is generic and independent of the media types of both contents and discussion notes, we focus on its special application of discussions around lecture recordings. The tool that this work builds on and extends is the “Authoring on the Fly” (AOF)
system [7] developed at the University of Freiburg. After relating the concept of discussion to collaborative annotation of digital media, we list the requirements of a general annotation and discussion system and present a data model for annotations and the architecture of the Annotation Web Service (AWS) for shared annotation of documents. A number of related works are examined with their advantages and shortcomings, before we describe the implementation of a web service for anchored discussion of lecture recordings.

DISCUSSION AS COLLABORATIVE ANNOTATION

Online discussion forums are a widespread means to integrate collaboration into computer-supported instruction. Net-based discussion can be seen as sharing notes which – explicitly or implicitly – refer to a previous note or to some part of the artifact(s) under discussion (or both). Thus, it makes sense to view online discussion as collaborative annotation of documents, where an annotation is a note with an explicit anchor to the content it annotates.

Suthers [11] distinguishes between different arrangements of the discussion notes and the artifact serving as an anchor for the annotations. Each of them has its own strengths and weaknesses. In the parallel arrangement, the artifact and the discourse are physically separated in two different applications or windows. Typically this layout is useful when holistic annotations are given to static or continuous documents. However, users have to switch back and forth between the two. Usually, learning management systems represent forums and learning objects in this parallel layout. An example would be a lecture recording watched in a special player and a web forum in a learning management system for discussion displayed in a web browser, where users have to constantly move from one medium to the other. This split attention is likely to increase cognitive load [12] on the learner. Also, users have to make references to certain parts of the artifact explicit with linguistic means. Similarly, other participants have to find the anchors before understanding the discussion note. Also simply providing a space for discussion, however, does not necessarily lead to its use by students. It is necessary to consider the specific characteristics of the communication and collaboration tool and the representational system used for the contents when combining them.

If it is intended to give more specific feedback by annotating specific positions in the artifact the parallel layout is insufficient because references have to be described explicit by the users, which can lead to misunderstandings. These disadvantages can be avoided by embedding the discussion in the artifact. Here, discussion items are directly inserted in the artifact at the appropriate places. Examples are Wiki web pages that can be edited by users. A main disadvantage of this representation style is that the discussion contributions are distributed over the document. For the users the coherence of the notes is difficult to comprehend and following the whole discussion is more difficult.

A third approach, which combines the advantages of parallel and embedded arrangements, leaves the discussion separate from the artifact but provides fine-granular links to the respective reference points. Each annotation is anchored at a specific position in the document. Therefore, an annotation must store precise position information (e.g. $x$ and $y$ coordinates for static documents, and timestamps in the case of continuous documents), and the application displaying the contents must also be able to display the annotations. It was shown in the research field of learning protocols that this referencing feature is one key feature improving learners’ knowledge acquisition [10]. This linked representation of notes allows for flexible visualization and forms the basis of our discussion in the remainder of this paper.

REQUIREMENTS OF AN ANNOTATION SYSTEM FOR ANCHORED DISCUSSION

This section summarizes the most important features that a general annotation and discussion system should provide.

Different document and annotation types should be supported. Many systems are tailored towards annotating one specific type of a document, for example web pages. Regarding the contents of annotations, the user typically is also restricted to a particular document format and modality (usually typewriting the comments). In order to offer more flexibility, the system should be expandable, so that any media types (such as graphical or spoken annotations) and document formats can be handled.

Different views of the annotations should be provided by the same application in order to support different learning and working styles. In particular, it should be possible to maintain the learning context while reading the annotations and to avoid media discontinuities. On the other side, different platforms may have certain constraints which have to be considered: For example, on a handheld device, annotations may presented in a work space separate from the resource for reasons of screen real estate. The document may have only a few visible hints (e.g. icons) on the document which can be used to access the corresponding annotations.

The anchor of an annotation can be a document or another annotation. This feature allows the use of the annotation system for threaded discussion. It may also make sense to allow references to both a previous note and a document, so learners may reply to a discussion note and in their reply refer to a part of the document under discussion.

The granularity and precision of the anchor should be as high as possible, because a precise anchoring of the annotation in the artifact is important for the understanding of the document context. It depends on the type of document to be annotated and on the application used for displaying the document. For example, unknown documents can often only be annotated as a whole, whereas in a text document, fine-granular references to individual words or even characters are possible. Therefore, the abstract description of an anchor should be flexible enough to represent different kinds of anchors. An application for annotating HTML- or XML-pages may use the structural information of the page in order to define an anchor. The annotation may, for example, reference a paragraph or any markup unit of the document. A video annotation system, on the other hand, annexes an annotation to
a video segment (i.e., a time interval in the video) or to a certain video frame (point in time of the video).

Authentication and access control. Learners use the annotation and discussion features for different purposes and in different learning phases. While reading or replaying documents (e.g., lecture recordings) for the first time, a student may use comments to clarify unclear sections of the document. The student will create private notes to memorize and identify important parts of the documents. Only a few of the private notes are usually shared with fellow students. Usually a student is a member of different learning groups: Private learning groups with friends and/or an officially assigned tutored group. Therefore, the system must support different discussion spaces managed by different people. The public discussion space is administered by the lecturer and the official learning groups are coordinated by tutors. Private learning groups can be created by the students themselves. The creator of such a group is responsible for the group and can invite others to join the group. The private annotation workspace is of course managed personally by each student. An important question that has to be answered is where the private notes are stored. There are to two possibilities. The local storage on the private computer ensures that only the creator of the private notes can access them. On the other hand, access to these notes is limited. If users want to use alternative devices (such as PDAs, laptop, or other), they will have to copy all the annotations. A central client-server architecture for managing the annotations is a suitable alternative. All the annotations are stored separately from the resource documents. In order to publish private annotations the user only needs to change the access rights so a group or the public can read the comments. Only the creator has the right to edit or delete a note or change the access rights. On the other hand, this privilege should not always be granted, because a non-private annotation may already have been replied to by someone else. Deleting or editing such an annotation means to destroy its context. The original anchor of such an annotation would not be valid anymore. Therefore, if an answer to an annotation exists, no further modifications of the annotation should be allowed.

Search and filtering of annotations should be possible. Most annotation systems provide only limited search functions. We would like to support filter and search functions both on the client and the server side. Pre-selection on the server minimizes the data volume that has to be transferred to the client and can be very useful if the client, for example a PDA, cannot handle all annotations. On the client, further enhanced filter options can be provided. These features cannot be seen independently from each other. For example, the granularity of the anchor and the different views of an annotation depend on the document type or, to be more precise, on the document format. An application that annotates PDF texts and an application for annotating web pages will have different technical constraints and maybe also different design objectives. Nevertheless, the abstract data model for an annotation can be defined independently.

The proposed data model allows the description of an annotation independently from the document type and format. Thus, both the annotated document and the contents of an annotation could be in any format. Different types of information are associated with an annotation (see Figure 1).

**FIGURE 1**

**INFORMATION ASSOCIATED WITH AN ANNOTATION.**

The **scope** defines which users are allowed to read the annotation. The values may be private, public, or a group. The **ID** consists of a URI (Uniform Resource Identifier) or a unique number in the database identifying the annotation. Each annotation can have a **subject**, or a title. This is well-known from web-based discussion forums, where the subject line is often used to represent the note in a collapsed view of the discussion. An additional item stores the user name of the author of the annotation, which is also required to manage access rights.

Each annotation has two anchors. The **context anchor** describes the annotated resource and includes three entries: first, a link (URI) to the external resource if the annotation is not embedded in the resource itself; second, a **data field** describing the exact position in the resource (in a format-specific way); and third, the field **format** contains a “mime-type” entry, so the application knows how the information of the data field has to be processed.

The second anchor is the **content anchor**. It describes the content of the annotation. For example, the text comment of a PostIt is stored in this entry. The structure of this anchor is similar to that of the context anchor: it contains a link to an external resource if the content of the annotation is not embedded directly in the annotation. The **data field** describes the exact position in the external resource (if applicable). Otherwise, the data entry contains the actual content, for example, the text comment of a discussion note. Again, the **format** field contains the appropriate “mime-type” entry. For example, a text comment could be in HTML or in plain ASCII.
In order to support anchored discussion, two additional fields are required. One entry references all the annotations made in answer of this annotation. And another URI identifies the annotation that this annotation is a reply to. Thus, each note can be a reply to exactly one other note, but multiple answers or comments to each note are possible. This conforms to the standard model of discussion threads.

Figure 2 illustrates the usage of anchors. In this example the original document (resource) has two annotations. Each annotation uses a context anchor to reference a position in the document. The positions in the document may differ. For example, the left annotation might represent a question asked by one student about a specific part of a lecture recording. The right annotation could be another student’s answer to the question. It is a reply to the first annotation and it references a possibly different part of the resource (maybe a section giving a clue to the answer). The content of this reply is an external resource, maybe a web page or a different lecture recording containing the answer to the question. Since we use URIs in the context anchors, we can store the annotations separately from the resource. The resources themselves can be received from the web, from a CD-ROM or the local file system. Any document, e.g. text or video, can be annotated. With this model it is possible to design a threaded discussion forum. In the example the links between the two annotations show that the right annotation is a reply to the left annotation.

![Figure 2](example_of_annotating_a_resource_using_the_context_and_content_anchors.png)

We adapted this idea of context and content anchors from the Common Annotation Framework (CAF) [2] in order to provide threaded discussions and to position the anchor precisely in the document and the annotations. Annotations are stored in an XML database in a lightweight XML description: The described elements including the element structure (see Figure 1) are mapped one-to-one to corresponding XML elements.

### RELATED WORK

Anchored discussion of lecture recordings has been implemented in several systems. In the MRAS system [1], the discussion notes are anchored at a time-stamp of the lecture video and are displayed only in separate frames next to the contents. Thus, a precise spatial anchoring is not possible. Another shortcoming is a time delay of about 10-15 seconds between the relevant content and the notes, resulting from the fact that learners cannot comfortably navigate back to the exact time in the streamed video after they discover a relevant topic to contribute.

The eClass project extended their lecture recordings with the CoWeb [9], which is based on the Wiki principle. At the end of each slide of a recording, learners can add notes which can be edited by every user. A slide as an anchor for the annotations is very imprecise and of rough granularity.

With the system described in [3] it is possible to annotate video streams, but only with temporal anchors as well. The video sharing and annotation system VSA [4] allows users to attach notes to streaming video. They are anchored only with an inexact parameter of the timeline. Among other things, teachers using VSA demanded more precision for the point of attachment between video and annotation, context dependency of annotations on the active video segment and video skimming capabilities for previewing video. With all these systems it is not possible to make spatial references, e.g., to a certain part of a slide.

### DISCUSSION BASED ON LECTURE RECORDINGS

We have implemented the generic annotation model as a web service where the clients are distributed instances of aofJSync, a software player for lecture recordings produced by the AOF (Ref) method. Discussions are modeled, as outlined above, as a collection of annotations belonging to the same document.

In general, the usage scenario is as follows: a learner downloads the e-lecture from a repository over the internet via HTTP or FTP (see Figure 3) or from other storage media (such as CD-ROM) and opens it with aofJSync. This player allows replay and annotation of lecture recordings. The client offers the usual navigation features for starting, stopping and pausing replay known from media players. In addition, it enhances navigation in lecture recordings through a table of contents with thumbnails linking to the first occurrence of the respective slide, buttons for slide-wise skipping (back and forth), adjustment of speed for time-scale modified playback, and visible scrolling. These features, especially the last one, allow for very fine-granular positioning in the document, which is also important for precise anchoring of annotations. Suppose a learner viewing a recording gets to a part of the lecture that is unclear to him or her. With a simple mouse click, the student can create a note and type in a question or comment. This note will automatically be anchored at the position of the mouse click and at the current time-stamp of the recording. Easy spatial and temporal positioning avoids the main disadvantages of most other systems described above. The creator of the note can set the duration of visibility with a slider (see Figure 5). By default, a note will disappear when the next slide is shown. The scope of the note (public, private, or group) can be changed, e.g. to make an originally private note available to other co-learners. Sharing notes is...
accomplished by synchronizing the client with the server. On the technical level, all requests are transmitted via the SOAP protocol\(^1\) from the aofJSync client to the Annotation Web Service (see ② in Figure 3). If there are new annotations on the annotation server, they are sent to the client.

All learners logged onto the system can retrieve all annotations that are associated with their current lecture recording and that are visible to them. They can then create new annotations (for example, by replying to existing notes) and thus engage in discussion with their co-learners.

![Diagram of the Annotation Web Service](image)

**Figure 3** The Annotation Web Service

To support the learners in an optimal way it is very important to provide different views on the discussion notes. Each view has its strengths and weaknesses for certain situations.

In our implementation of aofJSync, two views are provided. The thread view shows the notes very much like it is known from common threaded discussion forums (see Figure 4, left). Replies are displayed as indented lines below the original note. When selecting a note, its content can be shown in a panel below the thread view.

The second view is document-centered: while a user is watching a lecture recording, the annotations automatically appear like a “sticky notes” posted directly on the slide at the appropriate time and place (see Figure 4, right). This is possible since each note is anchored both temporally (through the time interval specifying the duration of visibility) and spatially (through the x and y coordinates defining the relative position of the anchor on the two-dimensional slide). Alternatively, the notes may be minimized and displayed as icons or hidden altogether (e.g., if they cover too much of the original document).

With these two views, it is possible to follow the discussion thread note by note, or to follow it from the perspective of the document and view the notes anchored at specific regions of the artifact. Both views are interlinked: by double-clicking an annotation in the thread view, a user can navigate directly to the respective location of the document where that note is anchored and, for example, start replay from there.

![Diagram of Anchored Discussion of a Lecture Recording with aofJSync](image)

**Figure 4** Anchored Discussion of a Lecture Recording with aofJSync

Analogous to the two views, the user has also two possibilities to create new notes. A new annotation can be created directly at any position on a slide at any time in the document, as mentioned above. Alternatively, the learner can add or reply to a note in the threaded view of the discussion process. Since a reply usually refers to the same part of the artifact as the original note, its default document anchor is also the same. However, the user can change this anchor if the reply should refer to a different part of the document. Each note in the document view also contains a small thread view of its follow-up messages so users can see immediately if there are replies and look at them (see Figure 5).

![Diagram of Visualization of a Single Annotation](image)

**Figure 5** Visualization of a Single Annotation

---

\(^1\) http://www.w3.org/TR/soap/
At the moment, two types of annotation contents are possible, purely textual notes or URLs of web pages. (In the latter case, the respective HTML document is shown in a mini web browser inside the “sticky note”.) In addition, we plan to integrate graphical annotations made directly on the slides and spoken comments (and combinations of the two).

CONCLUSION

The automatic capture of live presentations is used extensively at universities and other institutions. A drawback of these lecture recordings is their limitations in the communication facilities. To overcome this obstacle we have introduced collaborative annotations as a way to promote online discussion of multimedia lectures. Different document and annotation types, different views and anchoring of annotations are general key features of an annotation system for online discussion. On basis of these requirements we have deduced a generic data model for annotations which can be characterized in particular by a context and a content anchor.

Thereafter we have discussed several systems supporting the discussion of lecture recordings, but none of these systems allows spatial references to a part of the document and the precise definition of the specific time-anchor of an annotation. These disadvantages have been resolved in the Annotation Web Service, which was implemented as part of the Authoring on the Fly system. While viewing a lecture document, learners can use a slider to visibly scroll to the exact point of the lecture where they want to add a note. By clicking directly on the associated contents, a new annotation can be created which is anchored both temporally and spatially in the document. The duration of visibility for each note can be set by the user who created the note. The scope of a note can be declared “public”, “private”, or “group.” While private notes are only visible to the person who authored them, public and group notes are shared with other learners, who see the notes both in a thread view and in a document-centered view. These two views are linked such that clicking on a note in the overview will immediately navigate the multimedia document to the referenced position.

In our current work we are supporting the collaboration around lecture recordings with cooperation scripts which is a method to structure and to sequence online discussions.

ACKNOWLEDGMENT

Parts of this work have been supported by the German Research Foundation (DFG) within its special priority program on “Net-based knowledge communication in groups” and by the German Federal Ministry for Education and Research (BMBF) within the ULI project. Special thanks go to Daniel Eschke, who did most of the implementation.

REFERENCES