ARCHITECTURE FOR A VIDEOLECTURE ANNOTATION SYSTEM

Marco Ronchetti, Tiziano Lattisi, Andrea Zorzi

DISI, Università di Trento (ITALY)

Abstract

We present a system that allows students to take notes, which are synchronized with the lecture recording in the form of video. Such notes have the advantage of keeping a reference to the context where they were jotted, so that students can quickly review the portion of lecture where the annotation was taken. This also opens future possibilities of peer interaction such as annotation sharing, and of investigation of students’ learning behaviour.

Keywords: Video-lectures, note taking, multimedia annotation.

1 INTRODUCTION

The process of annotating text dates as far back as around 1000 AD when it became a prominent activity in Talmudic commentaries [1]. Annotations on the margin of a book even got a name in 1819, when the word marginalia was coined.

Not surprisingly hence, as soon as the web matured, research on the annotation of web pages became a research topic (see e.g. [2]). However it was not until recently that actually usable annotation frameworks gained popularity. They were allowed by the technology of the Web 2.0. Also, shared annotation of e-books has been recently explored (see e.g. [3]).

There seems to be almost universal agreement that note taking activity favors learning, probably since the student has to take an active attitude, rather than passively attempt to absorb knowledge. Our own research showed indications that students who actively take notes and also review them perform on the average better than those who do not [4].

Our motivation hence was to push students to take notes during lectures, and to enrich notes by helping students to link the notes themselves with the context in which they were taken: we intended to combine note-taking activity with the recording of video-lectures.

Recording traditional lectures in the form of videos is a practice that dates back to the end of the nineties (see [5, 6] for a review). Not only this practice becomes a substitution for students who cannot attend classes: videos can be used as ingredients for a flipped classroom approach (see e.g. [7]). When videos are available, students use them also to selectively review portions of lectures. In order to find the relevant portion of a lecture, tools and functionalities are needed. For instance, it is generally possible to seek a particular instant by using a time-bar. Semantic markers, such as indicator of slide transition, slide title or content also may facilitate search. To make this more effective, we investigated the possibility for a student to take (electronic) notes in class, and have them synchronized with the movie of the lecture being recorded in that moment. We believe that such a form of annotation, which ends up annotating a multimedia resource, could facilitate the task of reviewing particular portions of a lecture.

Hence, we developed a system that allows performing such operations. In this paper we will describe the requirements and the architecture of the system we developed. Experimentation in presently in progress, so validation of our idea will be reported in future elsewhere.

In section 2 we describe the background knowledge that sets the scene. Section 3 discusses the usage scenarios for which the system was developed. We then proceed with the description of the software architecture and of our prototype (section 4). The last section contains discussion and conclusions.

2 BACKGROUND

We started recording lectures in class long ago: we began experimenting in 2003, focusing on frontal lectures delivered with the support of presentation software such as PowerPoint™. At the beginning, we used a system called ePresence [8], which was designed to allow remote (and possibly deferred)
participation to scientific seminars. We decided to employ it in the context of frontal lectures, thinking that it could be a support for students unable to come to class (e.g. working students). To our surprise, we found that many students who actually had been in class used the system to review the lectures, or at least part of them [9]. For instance, they needed to check their notes, to review a portion of the lecture where they had difficulties or had lost concentration, or to clarify a point in a discussion with their peers when different persons had different understanding of a topic, and they wanted to check with the "oracle".

We understood then the need of some requirement for the software, such as navigability of the lectures, and the possibility to find a precise passage in a lecture. For instance, having thumbnails of the slides, searchable titles and text of the slides themselves, and ways to correlate them with a specific time in the video could help. We hence developed our own version of video-capturing software, called LODE, which evolved over the years. For instance, the better video resolution allowed by recent cameras and broader Internet band offers a much better user experience, and the possibility to wirelessly connect the acquisition system with the camera(s) facilitates the production process.

Videos (as opposed to simple audio registrations) have been considered useful mostly because they convey a sense of presence and familiarity with the teacher, which can help. Moreover, according to certain studies (e.g. [10]) the video favors maintain a higher concentration (maybe because it avoids being subject to incoherent external stimuli). The resolution available nowadays makes them even more powerful, since it allows using them as an extra sources of information, e.g. enabling the watcher to read notes drawn on the blackboard (which often complements PowerPoint™ presentations in an academic or scholastic environment), while in the past this was difficult due to resolution limits. Also, we started supporting the capture of whichever device is projected by the teacher on the class screen. This allows extending the lecture format, as not only the predefined PowerPoint™ slides can be captured, but also just about anything: from specific software demonstrations, such e.g. using Mathematica™, Matlab™ or a programming IDE to various types of multimedia (pictures or videos), web resources of any kind, and even handwriting (for instance we experimented using iPads™ as handwriting devices, e.g. instead of using blackboards for teaching mathematics).

The capturing process is seamless: we built a box that is put on the connection between the teacher’s presenting device (e.g. a PC or a tablet) and the screen projector: the box captures everything that passes on the cable, i.e. everything that students in class see on the projection screen. At the same time, the box captures the stream, which is wirelessly sent from the camera (audio + video). Post-processing software compresses the streams, prepares a web page and uploads everything on a dedicated the web site or on a learning management system (LMS). At this point the video-lecture is available on-line for students. (The post-processing takes a time of the order of magnitude of the lecture itself: typically a lecture of an hour can be published in about 15 minutes).

The web site (or LMS) can have open access, or be password protected, adjusting for teacher’s and/or institutional requirements.

Students need an HTML5 compliant browser (at present Google Chrome™ works well, while Mozilla Firefox™ and Apple Safari™ still have a few slight incompatibility with some feature defined by the W3C standard, which are requested by the pages our system generates). They see two video streams: one is produced by the camera and the other by the capturing of the screen projection. The two videos are always synchronized: when jumping backwards or ahead on one of them, the other maintains the synchronization. Earlier versions of our software were based - like many others – on the video stream plus static images (jpegs) showing the "current" slide. As we mentioned, this feature allows for an extension of the supported types of lectures, dropping the previous restriction, which required the lecture to be based on (PowerPoint™-like) presentation software.

Students have various control possibilities: they can decide where to put their attention, choosing the size of the streams: the can be equally sized, or one of them can occupy almost all the screen while the other is reduced to small dimensions. Watchers can use a time-bar to jump to different times in the videos. Pushbuttons for jumping ahead or backwards by small time amounts (10 seconds) are also provided. Since, as we mentioned, many students use the system for reviewing only a portion of the lecture, tools for quickly locating a particular time in the lecture are important: hence we provide also the possibility to play the videos at higher speed (1.1x and 1.3x). This feature is also used by students, who were not in class and assist at the lecture from home at their convenience: sometimes they are already familiar with the presented topic, yet they are afraid of losing some important information. In such case, they can use the high-speed reproduction feature to quickly overview the content, dropping at normal speed when they locate a section that is relevant for them. Also, an array of thumbnail of the
projection screen can be present, so as to allow to quickly locate the sought point of the lecture. If the lecture is based on PowerPoint™-based software, the thumbnail identify the slide transitions.

Students’ actions are tracked, and provide a rich body of data, which can be used to study students’ behavior and to find lecture characteristics, such as e.g. which parts of a lecture are most frequently viewed and which are just skimmed.

Figures 1 and 2 exemplify two of the discussed cases.

**KeyFrame**

Defines target values at a specified point in time for a set of variables that are interpolated along a Timeline.

The developer controls the interpolation of a set of variables for the interval between successive Keyframes by providing a target value and an interpolator associated with each variable. The variables are interpolated such that they will reach their target value at the specified time.

An onFinished function is invoked on each KeyFrame if one is provided.

**Figure 1.** Example of the visualization of a PowerPoint™-based lecture

**Figure 2.** Example of visualization of a math lecture based on an Apple iPad™ as replacement of the blackboard
Fig. 1 shows a classical PowerPoint™-based lecture, with the video and the slide, both of the same size. The white vertical bars on the dark and light blue horizontal band (the timeline) show the instants of the slide transitions. At the bottom the thumbnails of the slides are visible.

Fig. 2 shows a scenario where an Apple iPad™ is used as substitute for the blackboard. The teacher writes on the iPad™, the screen of which is projected in class and recorded in the stream. The visualization puts emphasis on the stream originated by the iPad™, offering a small view of the stream coming from the camera (of course this is only a choice, which the user can change at any time while watching the lecture.

Obviously the lecture model on which this is applied remains the classic frontal approach, which however is still by far the most common form of teaching in academic environment, especially when classes are large as typically happens in most bachelor courses, where the number of students easily exceeds hundred.

3 TAKING ANNOTATION WHEN LECTURES ARE BEING RECORDED: SCENARIOS

Even in the described scenario, students are not necessarily passively listening and absorbing knowledge. Most students actively take notes, but this part of the story has never (yet) entered into the lecture capturing process. We wanted to support this activity in the context of a lecture, which is being recorded and, later, watched.

Traditionally, students attempt to take notes about what the teacher says. When the blackboard is used, in general most students frantically copy whatever is being written on it. Often this activity even becomes primary, and prevents “understanding”, which is deferred to a later phase, when at home students read and/or reorder their notes. The advent of technology in the classroom has changed this custom. In the case of lectures supported by PowerPoint-like presentation software, often the slides are available (either before or after the lecture) on the course web-site or learning management system (LMS). If an Interactive Whiteboard (IWB) is used, whatever is written on it can be saved and later uploaded on the LMS or web site, freeing therefore students from the need of copying, and allowing them to use the time spent in classroom for understanding. Even in absence of IWBs, students can use technology: simply taking a picture of the blackboard with a smartphone can eliminate the need for manually copying.

Still, taking notes is useful. If for instance a copy of the presentation is available before the lecture, students can annotate it, e.g. by adding extra information that is given orally, contextual questions, outlining the most important points, etc. Even if the copy is only available after the lecture, the absence of transcription needs (because the student knows that the teacher’s material will soon be available) allows taking notes at a deeper cognitive level: they become critical observation, markings of particularly important facts or concepts, doubts. Often they are also taken to later trigger a memory recall of a relevant passage.

Can a system support such activities? Can it be friendly to both students, who need to use it, and teachers, who have to to fulfill the requirements needed by the system? We tried to answer these questions. What we wanted to achieve addresses multiple situations, articulated in time: first when assisting at the lecture, and later when studying. Let’s start with scenarios for first phase.

1 The student is in class. S/he has access (on-line, in electronic form) to the material that is being presented by the teacher.

2 The student is at home. S/he has not been attending the lecture in class, so s/he watches the recorded videos. While watching, s/he has also additional access (on-line, in electronic form, e.g. as pdf) to the material that was presented by the teacher.

In both these cases, we want to enable to student to (electronically) annotate the material. So far, this is not much different from what one can usually do: the material (e.g. the slides) can be downloaded and printed before the lecture, and then scribbled upon. It is also normally possible to jot notes on the electronic version of the notes, e.g. with a pdf annotator, or, best, on a tablet or tablet PC with an electronic pen. The main difference in our proposal is that the annotation happens on-line, because this will enable new functionalities when the student reviews his notes or the recorded lecture, as we will soon illustrate.
We have therefore to now discuss the second phase, which happens after the student has assisted (either in presence or virtually) at the lecture, and when he studies. We consider two cases: the first is when the study starts from reading one’s own notes, and the second when it begins from we-watching (a portion of) the lecture.

1. The student (who has already electronically annotated the lecture, either in scenario 1 or 2), reviews the annotated material. Hence s/he views (on-line) at her/his notes. S/he has now the possibility to focus on a specific note. At this point the system enables him/her to listen/view what was being said/done during the lecture at the time when the note was jotted.

2. The student (who has already electronically annotated the lecture, either in scenario 1 or 2), decides to re-watch the lecture, or a portion of it. When the time of the lecture comes, when the student took a note, the note appears. Also, the student has the possibility of navigating the video-lecture “by notes”: on the timeline a marker shows all the times when a note was added, so that the interesting spots can be easily reached.

There are then variants to scenarios 1 and 2: the teacher did not provide material. This may happen for a variety of reasons, e.g. the lecture is not based on material ready to be shown (e.g. in the “blackboard style” lectures, or in the case of live demos), or simply the teacher did not upload beforehand the slides s/he is using. In these cases, the system can provide a default on-line “blank notepad”, where the student can take notes, which will have all the properties of the notes taken in scenarios 1 and 2. Hence every annotation has a (hidden) time-stamp that allows correlating it with a time in the video.

4 SOFTWARE ARCHITECTURE OF THE SYSTEM

The key idea is to provide two web-apps. The first loads the material provided by the teacher (and if needed transforms it into a pdf document) and presents it to the student who is sitting in class (scenario 1). At this time, the lecture recording is obviously not available, since the lecture is being given A layer is superimposed over the pdf content, and the user can interact with such layer jotting notes. When a note is taken, it is saved both locally and on the web site. It is written into a database, with a time-stamp with the current time-of-the-day.

In parallel, the lecture is being recorded. At the beginning of the recording, a time-stamp with the current time-of-the-day is saved in the lecture metadata. In this way, it will later be possible to correlate every note with the right time of the video. Of course, it is needed that the clocks of the LODE recorder and of the students’ PC be synchronized: this is easily achieved by ensuring that all devices are on NNTP (the time synchronization protocol which gets the current time from the Internet).

In the scenario 2, the student is using the same app, but s/he is not in class: the lecture has already been recorded and the student is assisting to it in an asynchronous way, watching the video within the browser. S/he can control the video, pausing/restating and jumping forward/backward. When s/he desires, s/he can add annotations. Of course in this case the actual time-of-the-day does not play any role. The time-stamp for the annotation must be calculated by taking the time-stamp of the registration beginning, and adding to it the elapsed time of the video. In such way the notes of the asynchronous student are considered taken as if they had been added during the lecture.

Figure 3 shows the interface of the prototype we developed. Tools implemented in this app allow to draw lines, rectangles, circles and arrows. The color of the pencil can be chosen. Text present in the page can be highlighted. It is possible to type text onto the page, by choosing first the location on it where the text should appear. Extended notes with rich text formatting can be added in a pop-up window by means of a WYSIWYG (What You See Is What You Get) editor: once completed they are shown by placeholders (a symbol) shown in in the page. When mousing over the symbol, the pop-up window is revealed. A selector tool allows moving annotation in the page, and if desired to delete them. Of course the video component is only shown for asynchronous students, and is not present for students in class.

Obviously entering into this app requires the student to identify her/himself via an authentication process, which in turn requires first a user registration. Without identification it would be impossible to associate notes with a student.
Figure 3. View of the prototype app for scenario 1 and 2.

Scenario 1a and 2a (which apply when the teacher does not distribute in advance the lecture material) only differ for the fact that the annotatable pdf is just a set of blank pages.

In scenarios 3 and 4, students already assisted at the lecture (either physically or virtually) and took notes. Now they want to study, either starting by reviewing their notes (scenario 3) or by rewatching (a portion of) the lecture (scenario 4). Both scenarios are covered by a single app. Figure 4 shows the interface of the prototype of this app.

Figure 4. View of the prototype app for scenario 3 and 4.

The app shows (one of) the LODE stream(s) on the left hand side, and the annotated material on the right hand side. Which of the two streams (camera recording or screen recording) is shown is decided by the student, who can swap them at any time. The timeline component plays the same role it has in LODE, allowing to jump to different times in the lecture. Here it is enriched with markers, which show where (in time) annotations were taken, and which allow the user to quickly identify the location of the annotations. At the bottom, thumbnails show textual annotation: also these allow the student to navigate the lecture by notes.
The possibility to add notes after the lecture (as we discussed for asynchronous students) is available also to students who already took notes during the lecture: extra notes can always be added. Of course this is based on the same mechanism we described for asynchronous students.

Having discussed how the system offers functionalities, we can proceed with a technical overview of the system architecture.

It is obvious the system must be split in a front-end part, which is what the students see, and which lives in a web browser, and a back-end part, where the system logic and data are kept.

The front-end comprises two main environments: one for producing the annotations, a second for contextualizing annotations in a video-lecture.

The server side contains the engine of a web application and a database where annotations are saved. The video recordings and the notes provided by the teacher can reside on the same or (more generally) on other web servers.

In Fig. 5, the browser interacts with the main server (simply called “Server” in the picture) to load the web application. The server refers to the LODE component to deliver the video-lectures, and the browser gets them for this component. It also refers to the component called “PDF Server” to get the teacher notes (if available) or the blank notepad. It refers to the DB to save and read notes content and meta-information, while the course site (Sito del corso) is an entry point where user authentication and authorization is performed.

![Figure 5. Logical architecture of the annotation system](image)

The paradigm we chose was the Single Page Application (SPA), implementing two web apps: the page where notes are taken, and the page where an annotated lecture can be reviewed.

Adopted technologies include NodeJS for the server, and Angular2 as framework for the client side. All the code has been written in Typescript, an evolutionary version of JavaScript. For the database we chose no-SQL technology, and employed MongoDB.

The interaction between the SPA running in the browser and the server happens via WebSockets and relies on a RESTful API we developed. The API covers three major functionalities: user registration/authentication and credential management, management of the pdf documents, integration of the annotations with the LODE player.

The SPA needs to interact with the server, but also to retrieve the original pdf, the videos and their data: all this may reside on different web servers, which raises the issue of Cross Origin Requests: a practice which is forbidden, unless it is dealt with by using a Cross Origin Resource Sharing strategy. Such solution however would have implied severe complications: we therefore preferred to give the server also another task, i.e. to be a caching proxy for the pdf documents and for the video stream data (direct access to the video streams was not a problem, but their accompanying data suffered of the same Cross Origin problem).
5 DISCUSSION AND CONCLUSIONS

The work we presented introduces an innovative idea: to connect students’ annotation to the context in which the note was taken. This is achieved by creating a mechanism that links annotations (which can be drawing, handwriting, outlining or writing short or long text) to a particular time in the recording of a lecture. Vice versa, a video-lecture exhibits links to annotations. When watching a (portion of a) video-lecture, every student sees his own annotations and can navigate the lecture by using them.

This feature can serve multiple goals, such as e.g.:

- Notes can be taken as usual. Sometimes though after taking a note we forget details about its context. Having notes linked to the video can help reconstructing the context.
- Notes can be reminders, e.g. that a passage in the lecture is particularly difficult and need to be reviewed.
- Notes can become metadata for the video, becoming markers to identify sections and contain a short description of the section itself. For instance, during a math lecture notes can delimitate the beginning and the end of the demonstration of Rolle’s theorem.

Although we envision examples of such possible usages, we need to put the system into production to verify that our intuition is actually considered useful by students, and if so to identify the ways in which students actually use them.

Also, new scenarios open up: at present notes are considered as individual assets, but nothing prevents from using them as collective resources: a student could share her/his own notes with friends (e.g. over Facebook) or with the class. Also, this could allow the teacher to view students’ notes, either in detail or in a statistical report (e.g. showing the density of annotation in various passages of a lecture). Of course, questions about usefulness and user acceptance of such scenarios deserve to become matter of investigation.

This paper hence opens new potential research threads that, aside enquiring how user will appreciate and employ these new features, should necessarily include also investigations about the effectiveness of the user interface of the prototypes, and how it can be made better. All these issues are in our plans for the future.

ACKNOWLEDGEMENTS

This work has been performed in the context of the MIUR project CTN01_00034_393801 “Città Educante”.

REFERENCES


