

An Authoring Tool for As-in-Class E-lectures in E-learning Systems

Nael Hirzallah

Computer Center, Applied Science Private University, Shafa Badran, Amman 11931, Jordan

Abstract: This study lists and discusses the basic features that any E-Learning system should consider. It also proposes an authoring algorithm, namely Slide Generating Algorithm that uses the Authoring-on-the-Fly approach to create electronic course content from real lectures. A tool using the proposed algorithm was implemented packaging the outcome in a SMIL format including audio, video, and slides considering lecturer notes, animations and significant slide transitions. This tool allows instructors to easily create their E-Lectures during a normal lecture time. Furthermore, the outcome of the tool could be uploaded by any platform and viewed by any system.

Key words: E-learning, authoring tool, e-lectures, multimedia, eContent, SMIL

INTRODUCTION

While E-Learning systems are becoming widely spread in educational institutes, many such institutes are focusing on providing electronic course content (E-Lectures) besides various services to their students, staff, and employees. Most of the burden in offering these services such as communications, e-registration, and online exams is in getting the right platform and securing its database communications; whereas, the main burden in offering E-Lectures is in composing the content in an electronic format that could be uploaded through the adopted platform, viewed by all students, and easily composed.

The major three problems that may exist in the way of fully utilizing the E-Lectures feature are: the format needed may be specific to the platform; thus, migrating to a different system in the future would be costly on the institute when a good number of such E-Lectures have been already composed. Second, instructors may complain about how time consuming is to learn and compose a short lecture in the required format. Finally, the students may find themselves obligated to download some players to be able to view the E-Lectures. Thus having the top platform and tools available without the necessary content is just like having a state-of-the-art computer screen without a computer.

In what follows, the paper will categorize a number of essential E-Learning system services and features into five modules namely: E-Lectures, grading, assignments, communications, and supervision; claiming that these would compose a list of the minimum requirements that any E-Learning platform

must provide. The paper then presents an authoring tool that automatically creates E-Lectures through the proposed Slide Generation Algorithm. This algorithm intelligently captures screen shots attempting to record significant slide transitioning, notes and animations. The tool packages the lecture presented to the students in the lecture-room as is; with video, audio and animations using a standard format, namely SMIL SMIL^[1], that may be viewed by popular and publicly available players. But first, let's have a look at some available and related work in both the literature and the market to E-Learning systems and E-Lectures authoring tools, respectively.

An Overview on e-learning systems: E-Learning, according to Derek Stockley Stockley^[2], is the delivery of a learning, training or education program by electronic means; involves the use of a computer or electronic device in some way to provide training, educational or learning material. However, many argue that the communication between trainers and trainees or instructors and students is an essential part of the E-Learning process. Although we don't truly argue against this claim, but among them are those who go an extra step in claiming that having online meeting software might be enough to grant the institution an E-Learning provider title. This, however, we clearly argue against through our presentation to the number of services that could represent the minimum requirements of any E-Learning platform.

Tools such as Blackboard, WebCT and so on 'encourage student collaboration; improve team working skill and independent thinking'^[3]. Blackboard

software which is available in the market, is used to manage E-Learning (Blackboard Academic Suite), transaction processing and e-commerce (Blackboard Commerce Suite), and online communities. Among the features that the Blackboard Academic Suite offers, which is characterized to be flexible and extensible, is to provide the instructors with the ability to create and manage course content, utilize publisher content, evaluate performance, and communicate with students.

Although Blackboard software is not an Open Source system, but the company provides an open architecture, called Building Blocks that can be used to extend the functionality of Blackboard products.

However, Blackboard is available at a price considered by many institutions as an expensive system to use. Furthermore, the content or lectures developed using such systems, in most cases, cannot be used elsewhere; thus, leaving the institution under a long term agreement with the vendors of these systems.

Moodle, Ref.^[4], is a software package for producing internet-based courses and web sites. It is an ongoing development project designed to support a social constructionist framework of education. Moodle is provided freely as Open Source software (under the GNU Public License). Thus, in short period of time, Moodle has reached to a state where many of the features wish lists, that most educational institutes have, are already being implemented or under development. Such features include: Question engine, Assignment types, Activity modules, Activity modules, and Course formats. Moodle is built using PHP technology, and can support many types of database (particularly MySQL).

However, there are some concerns about using PHP for enterprise applications. Large enterprises do not entertain PHP due to the lack of support or a throat to grab when things can't be made to work in a certain amount of time. The single driving force behind most companies' decisions to move forward with *any* platform, language, software, and hardware product boils down to support, accountability, and SLAs wrapped around these things. In fact, these companies spend more money on support agreements than they do on actual software.

Despite this fact, Moodle is becoming more and more popular. After all, it is free and allows users to enhance its feature set. Thus, the algorithm and the process proposed in this paper for E-Lecture creation could be easily implemented adding a thrust to its success and enlarging its circle of users.

An overview on authoring systems: There are several ways of electronic content creation for online courses. The most common way is to create the content of the

lecture for web-use by means of creating static or dynamic web pages with PHP or HTML. Other possibilities for content creation are through using specific application development, such as Java applets, or the use of multimedia production tools, such as Macromedia Flash^[5] and Multimedia Toolbox^[6]. The disadvantage of these methods is the extremely high cost for development.

To avoid such high costs, systems for the creation of E-Lecture during a normal lecture were developed. Thus, lightweight content production using presentation recording instead of traditional authoring systems has become an increasingly popular method of creating instructional media for offline use, since it makes use of already existing educational potential at universities. Teachers are normally well experienced in face-to-face education and training. Nowadays, live events, such as training sessions in companies or universities, are in many cases already based on some kind of electronic material, such as PowerPoint-slides, digital images, video clips, etc. The slides are used in the live-event, graphically annotated and verbally explained by the presenter. Hence, it is natural to exploit this effort already invested in live-events for the automatic production of instructional content for later off-line use as well.

This concept of content production by presentation recording was called "Authoring on the fly" (AOF). Its main objective is to automatically capture live lectures and presentations in classrooms and lecture halls, make them available as E-Lectures in a variety of output formats for access over the internet or on demand. The AOF concept, which started in the mid 90s, has been realized and implemented into several prototype systems over the last few years, such as TTT^[7] and AOF^[8]. The major advantage of these systems is the reduced costs for the creation of electronic course content.

However, using these systems, the viewers or students may find it difficult to access the electronic content or the outcome. This is due to the need of installing special programs to access the content. For example, in the case of TTT, both JAVA and JMF had to be installed. It is needed on the users' side, students and other interested viewers, who want to retrieve courses via the Internet, not be subjected to the torture of complicated installation and operating steps.

Many other research projects have explored this method, as well, such as the Classroom 2000 (eClass project at the Georgia Tech^[9]) or the Berkeley Lecture Browser^[10]. Today, a number of commercial products exist for this task, such as Camtasia^[11], Microsoft Producer, or LECTURNITY^[12], which is the

commercial implementation of the AOF project. These products make automatic presentation recording available to a wider, non-technical user community.

Another disadvantage that may arise using the available systems is the unsatisfactory tradeoff between the size of the resulted E-Lecture and the loss of significant animations and notations.

Minimum expectation of an e-learning system: One accepted standard to develop E-Learning systems is the IMS^[13] (Instructional Management Systems), a global learning consortium that develops and promotes the adoption of open technical specifications for interoperable learning technologies that become the standards for delivering learning products worldwide. A feature level survey was run by the computer center of the author's university. It involved close to 200 staff member during the first semester of the academic year 2006/2007, the most common services that an E-Learning system must provide to students, academic staff, registrar, students' sponsors, and administration, could be categorized in the modules shown in Fig. 1, which expresses a high level view to the modules illustrating the dependency relationships among them.

Each module is responsible for providing a set of features through its sub-modules that will be briefed in this section. e-courses module will be later discussed in a little more details in the following section:

Messaging module

- * Announcements: Allows messages posted by instructors, heads of departments (HOD's), deans, and the president to address the recipients within the domain of each one's privilege. Recipients could be either individuals, such as a student or a staff member, or groups of classes, departments or faculties.
- * Messages: Allows communications between students and their instructors, similar to an email system. However, the address book must be dynamically created through the course registration process.

Grades management

- * Gradebook: Stores student results and offers support for custom grading, grade weighting, item analysis and multiple gradebook views.
- * Gradebook Calculator: Calculates grades based on student activities and displays special students reports to instructors.
- * Workflow provides users with the ability to route content, such as class marks by instructors, to others for review or approval.

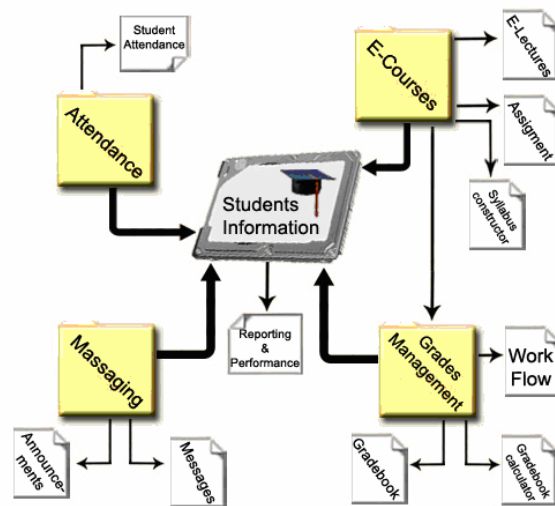


Fig. 1: Main E-Learning Modules

Students information

- * Reporting and Performance: Provides a view of student progress based on marks, warnings, and instructors feedback on them (similar to eBay seller/buyer positive and negative feedback), and indicates whether students have reviewed specific content items or not.

Attendance

- * Student attendance: Record students attendance at lectures, lab classes etc and then produce attendance reports.

E. e-courses

- * Assignments: It allows the instructors to create assignment items through which students can submit their responses. These assignments are only displayed during instructor-specified start and end dates and times. Thus, they can be viewed as Exams online as well.
- * Syllabus: It is mainly a tool to enter, update, and view a course syllabus.
- * E-Lectures: Through this, the instructors are given a tool to easily generate lectures that can be put online, and be viewable by students.

The Student Information Module is considered the central Module where most of the information is shared. For instance, the e-courses module extracts the students ID's who are registered in a specific course to allow them to enter and view both the assignments as well as the lectures posted. Whereas Syllabuses are available to whoever have access to the system.

Such system usually offers various services to various users depending on the role they would be playing. Eight differed roles could be offered. These roles are: Student, Instructor, HOD, Dean, Registrar,

Advisor, Embassy (or sponsor), and administrator. For instance, the administrator services include (but not limited to) accounts creation and manipulation, modules scheduling, Data manipulation, and backup tools; while Dean's services include monitor class occupancy, post announcements to faculty, review and approve grades, and view class communications statistics.

Proposed e-lecture authoring tool: Content creation for online courses may be time-consuming and costly process. Furthermore, should the course data, especially if in the field of computer science, change frequently, then the cost for the updates will add up. In this paper we also present a novel system using the Authoring on the fly approach, but the system itself includes two ideas. The first is about the algorithm presented for the Slide Generation, which takes into consideration significant animations and notations inserted by the presenter pre or during the lecture, while the second is summarized in the format of the resulted outcome of the lecture, namely SMIL, Synchronized Multimedia Integration Language^[1]. Choosing SMIL format allows an unrestricted use of all usual operating systems to view the content, and requires no complicated installation by viewers.

Nowadays, almost every live presentation is based on electronic material containing dynamic contents, such as animations and demonstration programs. A common format that is widely spread for slides is PowerPoint. Yet, authors may use dedicated standard tools for their lectures. However, we cannot expect that all data streams generated by running arbitrary applications can be captured in a symbolic, object-oriented form. Therefore, screen grabbing is the only reasonable universally applicable option for recording those streams.

Thus, our proposed algorithm, the Slide Generating (SG) Algorithm, captures the screen shots and intelligently chooses the appropriate key frames. These frames are supposed to have captured significant animations, slide transitioning and inserted notes. The system, which is based on this algorithm, as illustrated in Fig. 2, combines these key frames with the Video and audio representing the presenter or instructor into a SMIL file format. In the following sections, we first explain the first part of the system, namely the SG algorithm, then we handles the output format, namely the SMIL format.

The proposed slide generating algorithm: The SG algorithm periodically grabs screen shots (or frames for simplicity). As depicted in Fig. 3, the algorithm then

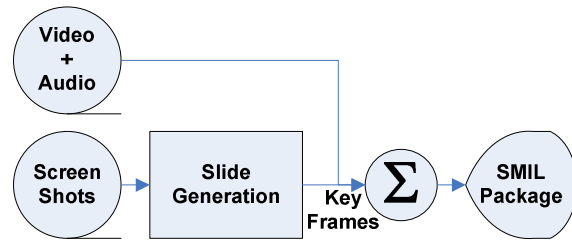


Fig. 2: System block diagram

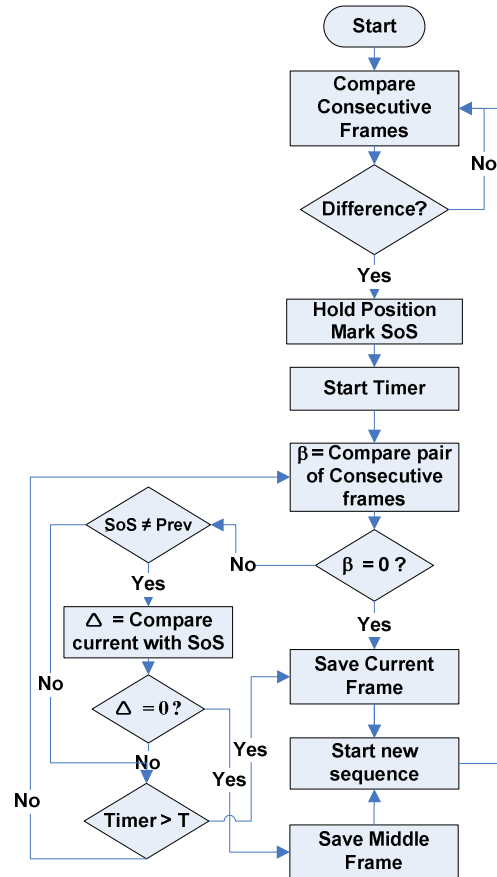


Fig. 3: Slide generation algorithm

compares the most recent frame, named by current frame, with the immediate previous frame.

Comparing frames could follow a basic approach to detect frame changes by comparing the intensity of corresponding pixels in the two frames. In simple words, if the number of pixels whose intensity values have changed from one frame to the other exceeds a certain threshold, a change is detected. However, since the type of frames considered here is different from that of normal video frames, because the frames are identical when there is no change, and basically no need to worry about brightness variations here, then a much

faster comparison could be performed through noting the JPEG file sizes of the frames. In comparing computer screen shots, a change in the JPEG file size would indicate a change in the frame. This way, the comparison process is much faster.

If a change is detected in the current frame, then the algorithm marks a start of a change sequence (or simply SoS), which may represent some animation, slide transitioning, instructor notes being inserted, or simply a new slide. Starting from this frame, marked as SoS, two kinds of comparisons will then be performed per each current frame.

As illustrated by Fig. 4, the first comparison is performed with the immediate previous frame (resulting in β values), and another with the start of the current sequence (resulting in Δ values). Furthermore, a timer starts counting the elapsed times till a decision been taken, as shown in Fig. 3.

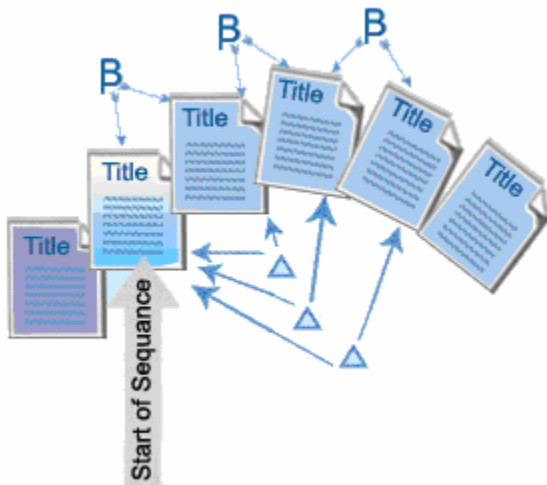


Fig. 4: A sequence of slides

The algorithm chooses not to save each frame upon any change detection for two folds. The first reason is to reduce the number of frames being saved per lecture. The second reason is due to the fact that frames during Slide Transitioning and notes insertion are usually insignificant for a lecture. Once the frames become stable with β equals to 0, that is, no more changes, the current frame will be saved. Otherwise, if the sequence duration becomes too long hitting a preset threshold equals to T seconds, the current frame will then be saved. This may indicate a significant change including animation and notes insertion, since slide transitioning and insignificant animations usually take short periods. Finally, when the sequence of changes ends up going back to the first frame or slide in the sequence, that is, the current frame becomes identical to the SoS frame;

this may indicate some kind of animation that is usually significant. Thus, the frame positioned in the middle of the sequence is being saved instead.

In summary, the algorithm takes the extra steps and performs multiple comparisons to eliminate insignificant changes, such as that caused by flying from left paragraphs, yet attempts to detect easy to miss significant changes such as gradual ones caused by animations or inserted notes.

E-Lectures format: As mentioned earlier, Some work has been done in the literature to generate lectures that can be put and viewed online, such as AOF^[14], and Classroom^[9], but the difficulty was having popular players be able to play back their outcomes. Initially, this presents two challenges. The first is to have an authoring tool that does not need special training, and the second is having the outcome format be played back by popular applications or Add-on's to popular internet browsers.

The proposed system adopts the SG algorithm then works on generating lectures in a common format that can be put and viewed online. This format is called SMIL. SMIL, Synchronized Multimedia Integration Language^[1], a W3C Recommendation, was our choice of the language to package an E-Lecture. SMIL allows integrating a set of independent multimedia objects into a synchronized multimedia presentation. SMIL, explained in^[1], although it is beyond the scope of this paper to be detailed, but as a quick introduction, it allows the authors to:

- * Describe the temporal behavior of the presentation,
- * Describe the layout of the presentation on a screen
- * Associate hyperlinks with media objects

Thus, by describing the regions that will compose the layout, then filling those with images, hyperlinks or video as needed, and finally defining a sequence of changes in one or more of these regions, the outcome would be a complete lecture. The following is a sample of the paragraphs composing a sequence. It specifies an image, named Slide1.JPG, to be displayed at a specific time for a specific duration in a specific region, and finally using a fade-in transitioning with the previous image.

```
<par dur="00:18" begin="00:15" id="par_01">  </par>
```



Fig. 5: An e-lecture sample

- * Table of contents as hyperlinks
- * Logo of the institution and copyright notes.

The easy approach to create such an E-Lecture was to combine the video and Audio captured by a camera and a microphone with a timed PowerPoint presentation slides saved as images and timestamps using SMIL. However, combining these manually is a time consuming process, which usually ends up using trial and error fashion to accurately synchronize the audio with images.

Furthermore, animations and instructors' on-the-fly notes will be lost upon manually saving the slides as images. On-the-fly notes are defined to be any high lightening, sketches, text additions, etc... entered by the presenter during the presentation. In PowerPoint, clicking the right-mouse button during the Slide Show mode and selecting the Pointer Options can access these.

Figure. 4 shows an E-Lecture being played by RealPlayer application (also available as a plug-in), a RealNetworks application. The students are offered to stream and view E-Lectures through their normal browsers within Add-on components publicly available. One of the advantages of such E-Lecture format is that it is hard to download the complete content. This is because the content will be distributed over multiple files, but referenced all by one SMIL file. This may prevent the E-Lectures from free distributions, unless of course, when it is explicitly offered in zipped files.



Fig. 6: Authoring tool

RESULTS AND DISCUSSION

The proposed SG algorithm was implemented in the authoring tool discussed in the previous section as a prototype and used by a number of staff members in both the IT faculty and Pharmaceutical faculty of the author university. Not much training was given to the instructors to use it but a single sheet on how to operate the tool. Each instructor used the tool to generate up to four 20 min E-Lectures. Although the tool could be used in class, but the instructors preferred to try it in their own space or offices. All users expressed their satisfaction at the outcome they got. Some insignificant slides were picked up such as that in Fig. 7, which represents a flying from top object at a very slow speed, yet saved by the system.

However, with minor adjustment to the given slider by the tool which refers to the outcome 'File Size' that could be set to 'Big', 'Small' or anywhere in between. This slider actually sets the 'T' value referred to by the algorithm expressed in Fig. 3, to 2 seconds if moved to

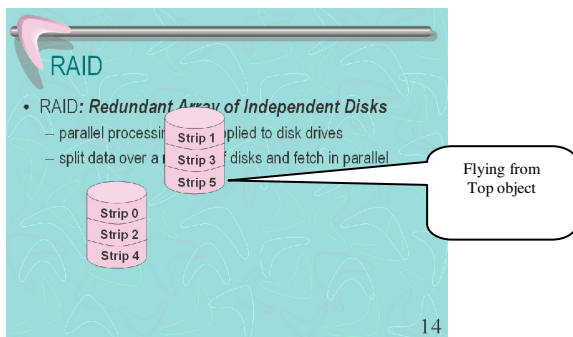


Fig. 7: Insignificant animation caught in middle

The layout sample illustrated in Fig. 5 is composed of four regions specified as follows:

- * Video showing the instructor in wide view or a close up view (including Audio)
- * Images showing the slides, animations, and instructor on-the-fly notes

most left position, 'Big', or 6 seconds if moved to most right position, 'Small'. Furthermore, these generated E-Lectures were shared by over 100 students and received no major complains since all were had public players anyways on their own systems like RealPlayer.

CONCLUSION

This study presented a list of features that would be essential for any E-Learning system to be called so. Furthermore, it proposed a Slide Generation algorithm that is features by the ability to generate slides from screen shots representing animations, on-the-fly notes, and significant slide transitions; giving some kind of live presentation or As-in-Class E-Lectures. The algorithm is being implemented in an authoring on the fly tool to create E-Lectures without the need for training or much knowledge in E-Learning tools and applications. Moreover, students would be able to get lectures while sitting at home through their web browsers without the need to install special applications, but their common ones.

REFERENCES

1. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, W3C Recommendation, 15-June-1998 (<http://www.w3.org/TR/REC-smil/>)
2. Stockley, D., 2006. E-learning definition and explanation (E-Learning, Online Training, Online Learning). <http://derekstockley.com.au/elearning-definition.html>
3. Border, J., K. Stoudt and M. Warnock, 2006. E-Learning Concepts and Techniques. Ch 4. Institute for Interactive Technologies. Bloomsburg University of Pennsylvania, USA, http://iit.bloomu.edu/Spring2006_eBook_files/index.htm
4. A complete web site on Moodle at http://docs.moodle.org/en/About_Moodle
5. Macromedia Flash <http://www.macromedia.com/>
6. Click2Learn Toolkitbook <http://www.asymetrix.com/>
7. Ziewer, P. and H. Seidl, 2002. Transparent Teleteaching. University of Trier.
8. Ottmann, Th. and R. Müller, 2000. The authoring on the fly"-system for automated recording and replay of (tele) presentations. ACM/Springer Multimedia Systems J., Special Issue on Multimedia Authoring and Presentation Techniques, 8: 3.
9. Gregory, D.A., 1999. Classroom 2000 - An Experiment with the Instrumentation of a Living Educational Environment. IBM Systems J., Special issue on Pervasive Computing, 38: 508-530.
10. <http://bmerc.berkeley.edu/frame/projects/lb/>
11. <http://www.techsmith.com/products/studio/default.asp>
12. <http://www.lecturnity.de>
13. IMS/GLC, Instructional Management Systems Global Learning Consortium. <http://www.imsglobal.org>.
14. Wolfgang, H., R. Mueller and T. Ottmann, 2004. The AOF method for production, use, and management of instructional media. In: Proc. of ICCE 2004, Intl. Conf. Computers in Education, Melbourne, Australia.