Enabling Active Engagement in E-tutelage Using Interactive Multimedia System

Xiaoping Wei, Haining Liu, Magda El Zarki
Information and Computer Science
University of California, Irvine
Irvine, CA 92697
{wei, haining, magda}@ics.uci.edu

Abstract

Computer and multimedia technology play an increasingly important role in education. However, current educational multimedia systems suffer from a major drawback that they don't support active engagement from users. In this paper we present a system that overcomes the hurdle. Our system offers rich user interactivity; it naturally supports and invites user involvement in a multimedia presentation. In the paper we first categorize the different levels of user interactivity that we deem vital for an interactive multimedia system, and then we describe a system design that enables active engagement.

1. Introduction

Computers have become such an indispensable part in our daily lives that they have gradually affected the very way we teach and learn. During the past decades, the basic model of education delivery has been dramatically changed with the advancement of computing technology: from the traditional learning model which includes direct interaction between lecturers and audiences, passive learning, private reading and introspection to computer-aided distance teaching and learning. More often, the human teachers are completely substituted with computer systems. For example, any registered IEEE member can take the Web-based on-line courses provided by IEEE that are totally instructed by computers.

Nowadays, a typical on-line course might just require a student with basic computer operating skills to be seated before a computer, and the Web-augmented or Web-delivered electronic content is simply several clicks away. The whole learning process is different from the traditional way in that it is asynchronous, more private and self-paced. Moreover, the delivered electronic content is moving quickly towards multimedia--text, graphics, sound, animation, and even full-motion video. The pupil can then access this richer content in his/her preferred order, at any time that is appropriate. Access to course material becomes individual; the student receives the tutoring when one needs it most, in the sequence and quantity for which one has prepared [1]. It is foreseen that such an "any where any time" paradigm will replace the traditional education model eventually, and we will finally step into an "e-tutelage"[1] era.

However, despite all the benefits the "e-tutelage" has brought to us, it also raises new challenges to the educators. Experiments show that the actual effect of Web-based distance learning is still not satisfactory. "The majority of prepackaged multimedia education products are educationally poor" [2]. The reason is rather straightforward: current "e-tutelage" techniques overlook the very essential element of a successful learning process--active engagement. While the great advantage of the "e-tutelage" model itself is that the pupil is actively engaged in the learning process, actual live engagement using current technology is not yet available. We describe below two teaching applications in which active engagement would vastly enhance the service:

Teaching children with Attention Deficit Hyperactivity Disorder (ADHD). Children diagnosed with ADHD are very challenging to teach due to their highly fluctuating energy levels. The ideal teaching tool would be one that allows the parents/teachers to tailor the lesson content to fit the child's specific learning styles and abilities and also allows the child to interact with the presented material. In other words, the child can manipulate the different objects contained in the presentation (jump forward, go back, enlarge, remove etc.), enabling it to progress at its own pace. This requires that the content be object based and accessible/changeable in real-time by the user.

Foreign language learning. Learning a foreign language can be a daunting experience for some of us. The ideal learning environment would be one in which the student could 1) listen to a lesson in the new language as well as the native language, alternating between them, 2) that each video/audio track be accompanied by subtitles in the other language, 3) have the ability to FF or rewind at any point in the lesson and 4) have an accompanying text window that provides phonetics, grammar rules, etc related to the material being covered. Current distance learning technology does not support high levels of user interactivity for remote access, i.e. most of these interactive features are disabled.

These two examples show us that current "e-tutelage" techniques simply deliver content with no recourse to allowing students to actively engage in the learning process [2]. The lecture content usually cannot be reordered or segmented from its original state during the delivery process. Given the fact that a successful learning experience basically lies in interaction--the student must actively participate in the learning, it is imperative that we work to enhance the process with more interactivity support.

2. Interactivity

In this paper we model e-tutelage as a process which has 3 major parts as depicted in figure 1. We will use a distance language learning application as an example throughout the rest of the paper to illustrate our system design.

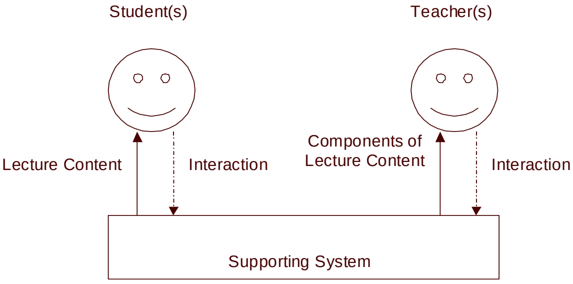

Figure 1. A model of e-tutelage

Both the teacher(s) and the student(s) interact with the supporting system, and they interact with each other via the supporting system. Besides the expectations we have identified above, it is also highly desirable that the teaching content can be prepared using multiple components. In other words, the teacher(s) can assemble a lesson based on all the available pre-recorded components for the targeted student(s) easily. Ideally, such a composition process should mainly be associating the components spatially and temporally, and the supporting system only needs to store a description of this presentation. Once the learning process, i.e. content delivery process from the supporting system to the student(s), is triggered by the student(s), the supporting system starts pumping the final presentation content on the fly. During the playback of the presentation, the student(s) can interact with each component or a group of components. For example, in a foreign language learning process, the student(s) may want to slow down or skip certain parts of the presentation, manipulate the volume of different languages, or turn off one language completely.

Apparently, different kinds of interactions are involved in such an e-tutelage process from both sides. To fully enable these interactions, we further classify them into three categories, namely, presentation level interactivity, session level interactivity and local level interactivity [11]. The definitions are detailed as follows:

- Presentation level interactivity: in which a teacher/student makes changes to the scene by changing the characteristics of a presentation component (on/off status, size, resolution, sound, language track, etc.).

- Session level interactivity: in which a student controls the playback process of the presentation (i.e., VCR-like control for the whole session and for components (or group of components if temporally linked).

- Local level interactivity: in which a student only makes changes that can be taken care of locally, e.g., changing the position of a component on the screen, volume control, etc.

In this paper we present a client-server based supporting system that encompasses the above 3 levels of user interactivities to improve e-tutelage. Our design is motivated but not limited by MPEG-4, a new ISO/IEC standard [3][4][5]. With MPEG-4 technique, the final multimedia presentation is treated as a superset of media objects, which can be natural/synthetic audio or video, text and graphics. Each object is carried by one or several elementary data streams, and delivered to the end user via different transport channels. By doing this, the end users can interact with the multimedia presentation and manipulate the different media objects. For example, end users can change the spatial-temporal relationships for different objects, turn on or shut down media objects, specify different perceptual quality requirements for different media objects, and have VCR-like control over the session or object groups. It then becomes possible to enforce the active involvement of end users. The system design is presented in the following parts.

3. Authoring Tool and iPlayer

The three levels of user interactivities are embedded in two application-level components in our system: authoring tool and interactive player (iPlayer).
Our authoring tool enables a user to compose a presentation tailored to one's interests or preferences, therefore the authoring tool itself reflects presentation level user interactivity. In contrast with most existing authoring tools which are typically coupled with encoder to produce a file storing all the contents of a presentation [6][7][8][9], our authoring tool is completely independent of encoding process, and simply generate a SMIL-based [10] presentation description (PD) without any real media data in it. Rather, the media contents can reside at different places from a user's site, for example on a remote server's data repository; they are retrieved and streamed to the user when the presentation is requested. The PD contains a scene description (SD) that indicates the temporal and spatial relationships among objects; it also includes an object descriptor for each media object (OD) that specifies various attributes of the object.

As an example, suppose we author a distance language learning presentation as presented in figure 2. The presentation consists of a group of objects: a 3D room, a background picture, some furniture items, and a person that is composed of three parts: a video, an audio and a subtitle revealing what the person is saying (this could be further enhanced by the addition of a power point slide presentation with relevant content, for example the pronunciation or usage of a word, a picture related to the speech, etc.). The PD for this presentation is illustrated in Figure 3.

Figure 2. A sample presentation of distance learning.

The scene description of a PD has a tree structure describing how a presentation can be decomposed into individual objects. The inner nodes (dotted circle) of the tree structure represent virtual objects and the leaf nodes (solid circle) denote real media objects. The scene description here shows that the person's video, audio, and the text subtitle are synchronized with each other. The scene description specifies temporal information for each object such as its starting time and duration. The scene description also contains spatial information for each object that indicates where on the computer screen the object should be shown. The spatial information is not shown in figure 3 but is included in the resulting PD. Associated with each real media object is an object descriptor (rectangle) that lists possible values for the object's various attributes (bit rate, resolution etc.), the currently selected ones are underscored.

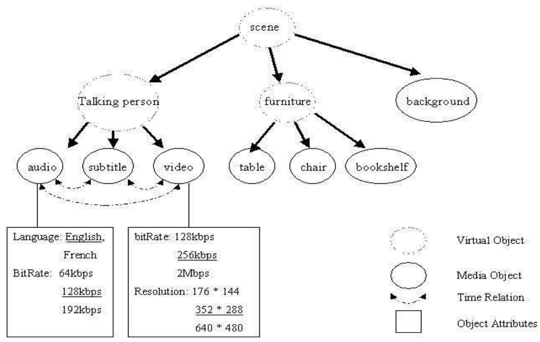

Figure 3. Presentation description of the distance learning example.

Our authoring tool features a web-based interface that allows remote (or local) access to media objects stored in the server's database. A user can browse the content, pick desired objects and drop them into the authoring window, choose the preferred quality (bit rate, resolution etc.) of each object, and specify the temporal and spatial relationship between objects. For the above example of distant language learning, the audio of the talking person and the subtitle text are recorded in multiple languages such as English, French, Spanish, Chinese, etc. The video is layer compressed into multiple video objects, one for the base layer and others for different enhancement layers. All the audio, video, subtitle text, and other related objects (furniture and background images) are stored in the server's media database. An instructor can then author a presentation with a particular video quality and a preferred language for the lecture. When a video object is right clicked, a window appears that lists all possible qualities (combinations of attributes such as bit rate and resolution) of the video and related language tracks and subtitles. For an English-speaking student who wants to learn French, it is desirable to have the voice speak in French and the subtitle text in English for translation. Also if the student has a high-speed (broadband) network connection, a high bit rate (or high resolution) video can be chosen. Authoring such a presentation with our tool is shown in Figure 4.

The authoring tool generates the PD for the presentation as illustrated in Figure 2. When a student/user wants to playback a presentation, either it is created by the student or user of the system and then the PD is sent to the server or if a pre-recorded presentation is requested, the PD already resides on the server and it just simple retrieved. The server then parses the PD and decides what media data to send and how to send them. The client also relies on the PD to synchronize media contents and render them on computer screen.

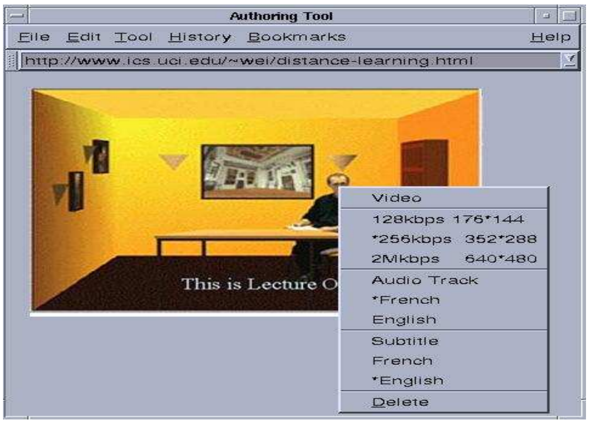

Figure 4: Authoring the presentation with our tool

The Interactive Player (iPlayer) of our system enables all three levels of user interactivity during the course of the presentation playback. In Figure 5 we show the iPlayer playing the distance learning language presentation mentioned above.

iPlayer consists of several buttons and sliders as shown, that allow a user to apply VCR-like control to individual media objects or to the presentation as a whole. A user can also change a media object's quality attributes or delete the object all together. For that a user right clicks the object, then a pop up menu similar to the one in the authoring tool opens up with possible options which are extracted from the object descriptor in the PD. Objects may also be added. When "Add" is selected, the authoring tool is fired up and the user proceeds to pick the required object, from the server's data repository, that is accessible to the user (e.g., a selection of objects that the teacher has made available as part of the course material such as the powerpoint slides that give the meanings/usage of words, a picture depicting the scene being voiced, etc.). The changes to the presentation will be sent to the server in an updated PD. The presentation proceeds with the new object added unless the user decides to restart the session.

Figure 5. iPlayer with the distance learning presentation

4. System Architecture

Our system puts a strong emphasis on the object-oriented structure to maximize the possibility of interactivity to the end users. Our system design is centered around this premise and built upon a client/server model. Figure 6 depicts the architecture of our system. It consists of 1) a server, which stores encoded media objects and serves streaming, 2) a client, which composes and/or playbacks the presentation, and 3) an IP network.

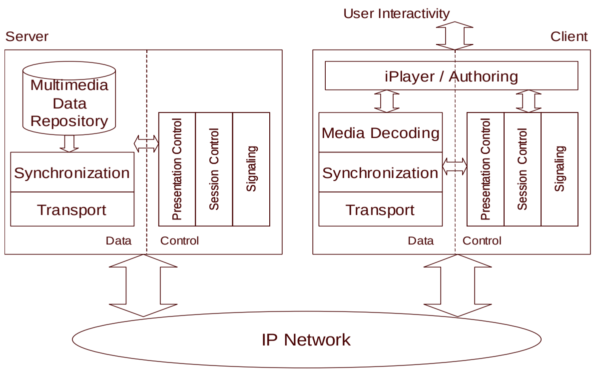

Figure 6. System architecture

The built-in functionality of the system can be vertically grouped into two planes: data plane and control plane. The data plane is responsible for all actions associated with the retrieval, streaming, and playback of the media data. The control plane is built for exchanging and processing control messages triggered by user interactivity.

Within the data plane, a repository stores all the media data, which can be used to compose the multimedia presentation, at the server side. State-of-the-art coding techniques, such as layered coding, arbitrary shape coding, etc. are used to prepare the data. This provides more flexibility in meeting the different demands for bandwidth needs and perceptual quality. For example, with arbitrary shape coding, the end users can easily mix and match foreground and background objects based on presentation needs, and better overall quality can be achieved compared with the conventional square frame rendering. The server associates the media objects with timing information on the fly according to the description of the presentation submitted by the end users. Different media objects are virtually "synchronized" after this. A particular synchronization module functions to achieve both the inter-object synchronization and intra-object synchronization. The synchronization module works tightly with the transport module and encapsulates the media data into timestamped packet streams. The packet streams are then delivered across the IP network to the client side. At the client side, the received packet streams are routed to the corresponding decoders. Finally, the decoded media objects are re-composed into a final presentation and that is then rendered to the end user. In our system, the transport modules at both sides work cooperatively to accommodate the fluctuating network bandwidth. The client-side transport module detects the network status and feeds it back to the server, and the transport module at the server side proactively schedules the delivery process, making the appropriate adaptations if necessary.

The control plane in our system interacts and cooperates with the data plane to support the three levels of user interactivity. Three distinct functional modules: presentation control, signaling and session control are built at both the server and the client side. The presentation description messages are conveyed between the presentation modules. These messages describe the composition of the presentation as tailored by the end user. The signaling modules are responsible for the delivery of the configuration information essential to session establishment, which includes media data type, encapsulation format, and transport channel identification (IP port number) etc. The session control module at the client side forwards to the server the control messages that result from session level user interactivities, i.e. VCR-like control. The server side session control module interprets the messages and maintains the state of the session.

5. Summary

In this paper we presented the design of a supporting system that enables active engagement for e-tutelage applications. This approach will result in a richer and more gratifying experience for the end user. The system gives control to the teacher on how best to compose a lecture presentation using a wide variety of tools and objects. The student/end user is given some freedom on how to learn the material, adding, changing or removing objects as necessary.

6. Acknowledgement

The authors would like to thank Liang Cheng, Anders Fridrikson and James Yeung of the Video over IP group at UCI, for their substantial contribution in building the prototype of the system discussed in this paper.

7. References

1. Marold, K.A. The 21st Century Learning Model: Electronic Tutelage Realized. *Journal of Information Technology Education*, Vol. 1 No.2, 2002. 113-123.
2. Chambers, M. The Efficacy and Ethics of Using Digital Multimedia for Educational Purposes. In Tait, A.& Mills, R.(eds): *The Convergence of Distance and Conventional Education: Patterns of Flexibility for the Individual Student*, London: Routledge, 5-16.
3. ISO/IEC JTC 1/SC 29/WG 11, Information technology-Coding of audio-visual objects, Part1: Systems (ISO/IEC 14496-1)," Oct. 2001.
4. ISO/IEC JTC 1/SC 29/WG 11, "Information technology-Coding of audio-visual objects, Part6: Delivery Multimedia Integration Framework (ISO/IEC 14496-6), Dec. 2000.
5. ISO/IEC JTC 1/SC 29/WG 11, "Information technology-Coding of audio-visual objects, Part8: Carriage of MPEG-4 contents over IP networks (ISO/IEC 14496-8), Jan. 2001.
6. MPEG4@ENST, http://www.comelec.enst.fr/~dufourd/mpeg-4/index.html
7. GSRT Authoring Tool, http://uranus.ee.auth.gr/ pened99/Demos/AuthoringTool/authoring_tool.html
8. Sung, M.Y. Rho, S. J. Jang J. H. A SMIL-based Multimedia Presentation Authoring System and Some Remarks on Future Extension of SMIL, *Packet Video 2002,* USA.
9. Shieh M.-J. Peng K.-L. Chen W.-C. The Design and Implementation of a Visual MPEG-4 Scene-Authoring Tool, *WEMP4 2001*, USA.
10. W3C, Synchronized Multimedia Integration Language (SMIL) 2.0 Specification, http://www.w3.org/TR/smil20/
11. Liu, H. Wei, X. El Zarki, M. A Transport Infrastructure Supporting Real Time Interactive MPEG-4 Client-Server Applications over IP Networks, *IWDC'01*, Italy