WhiteboardVCR: a Web Lecture Production Tool for Combining Human Narration and Text-to-Speech Synthesis

Ng S. T. Chong¹, Panrit Tosukhowong¹, and Masao Sakauchi²
¹Institute of Advanced Studies/United Nations University
²Institute of Industrial Science/University of Tokyo
chong@ias.unu.edu
panrit@ias.unu.edu
sakauchi@iis.u-tokyo.ac.jp

ABSTRACT
With rapid advances in computer speech technology, TTS (text-to-speech) synthesis is becoming increasingly attractive as a supplement or even as an alternative to the human narration. This paper explores the potential of TTS synthesis in Web lectures. We propose WhiteboardVCR, a new approach that we have developed for producing and presenting Web lectures. The system supports synchronization of slide markups and slide switching with a narration. The narration is a combination of video, human voice, and speech synthesis. Users can create a presentation on the fly or at leisure time, and edit it before publishing to the Web. Preliminary experimental results of using WhiteboardVCR from the perspective of the audience are discussed.

Keywords
Web lectures, Media synchronization, Text-to-speech synthesis, Distance education

Introduction
The Web lecture is becoming more and more popular as a technology for presenting course material on the Web. First, it is still the best replacement for real Web lectures. Neither textbooks nor handouts adequately replace an up-to-date lecture done by a leading researcher or professional. Second, Web lectures provide distance students with the “feeling of the classroom”. Some research claims that Web lectures are at least as efficient as regular lectures (LaRose, 1997). Third, recent years have witnessed improvements in media streaming technology. Streaming lets users view multimedia as it downloads. Many commercial software solutions now support the W3C multimedia standard SMIL (W3C SYMM Working Group, 1998) that allows multimedia objects to be synchronized in time and space. This has created a trend for slideshow style of presentations on the Web, where a sequence of slides is synchronized with a separate audio and/or video narration. A popular example of this type of media synchronization in streaming is RealNetworks (RealNetworks, Inc., 1995). Although streaming technology is gaining widespread use, it still faces many technical challenges. The most fundamental problem is its sensitivity to packet losses that can lead to content degradation (e.g., choppy video and audio).

In Chong, Tosukhowong and Sakauchi (2001), we presented the architecture and a prototype of a presentation tool that supports both media streaming and text-to-speech (TTS) synthesis as an alternative to audio streaming in synchronized slideshows. Users can annotate a slide and associate any part of an annotation with a written narration, which is then converted into speech at playback time. TTS offers numerous advantages over streaming audio, such as:

- Lower bandwidth requirements;
- There is no need for a streaming or special server other than a regular web server. Moreover, there is no need for any sophisticated acoustic arrangement to ensure good sound quality (e.g., to reduce background noise during recording). Hence, the production cost of a Web lecture could be significantly reduced;
- The speech transcript is readily available and can be used as redundant media that can enhance understanding, particularly for hearing-impaired users and users who are not proficient in the language of the presentation;
- Similarly, using the tool, a speaker who is not proficient in the language required for the presentation can communicate his/her ideas easily.
- If the transcripts of the presentation are available, the same presentation could be offered in different languages, without requiring additional recording sessions to generate the audio for each language.

However, based on our surveys some users felt that using only synthesized voice for presentation was not adequate in some cases. For example, without the naturalness of the human speech it would be difficult to emphasize important parts of the talk. The first version of WhiteboardVCR also lacks support for recording of
live presentations. Before the lecture can be delivered on the Web, long hours of synchronization and editing would be needed after the presentation, especially in the case that we already have the video and associated PowerPoint file. Thus, we have extended WhiteboardVCR with the capability to include synchronized audio/video streams in the presentation, which allows user to record their live presentations directly, and publish them to the Web without the post-presentation editing burden. We have retained all the features and benefits of using TTS technology in the new version. In addition, users have the choice to use synthetic voice only, human voice only, or a combination of both. This adds extra flexibility to our Web production system, as we can use our own voice for the main theme and TTS synthesis for the prelude, or vice versa. We will discuss some scenarios that could benefit from mixing human voice and TTS synthesis further in this paper.

Development History of the WhiteboardVCR Toolset

As part of a larger project at United Nations University, we developed First-Class, a customizable shared whiteboard in 1999. First-Class serves as a shared collaboration space for distributed users. It is itself a cooperative environment for both computing and integrating independently developed software components. This is true, in the sense that First-Class is an extensible collection of servers executing various services on behalf of remote clients (e.g., rendering math equations), and developers around the world can add new features to the system. Chong (2000) describes the details of this architecture. From the user's perspective, First-Class is more a medium than an application. It is an execution environment that allows users to dynamically import new features on demand at runtime. This is similar to the Web, which provides a runtime environment for applets. The access to a potentially unlimited number of features at runtime suggests that course materials presented by First-Class need not to be prepared in advance. Teachers could use our whiteboard to generate the content on the fly during the lecture presentation. First-Class supports both real-time and on-demand delivery of presentations, but it does not have any support for editing a pre-recorded presentation.

Our next project was WhiteboardVCR. Initially we aimed to create a presentation tool that would allow users to annotate slides and deliver a narration using synthesized voice instead of human voice. We chose to use the Text-to-Speech engine built for the Microsoft Agent, as similar features based on Java were not fully developed yet at that time. That led to the implementation problem of how we could establish communication between the WhiteboardVCR playback client and Microsoft Agent. We found that using a combination of Java, JavaScript, and Visual Basic script, we could circumvent the above problem for the playback tool. Note that the original version of WhiteboardVCR does not support the live recording of a lecture in progress. The normal usage scenario is to prepare a presentation beforehand and then publish it to the Web.

However, there are cases when presenters do not wish to present using TTS altogether. This is in part due to the fact that using pure TTS voice for the entire presentation can fail to keep the audience motivated and impressed. Presenters who have good command in the language and presentation skills may need only annotation functions or TTS synthesis occasionally to emphasize the presentation. In that case, the lecture is mainly communicated using human voice. The next step in the evolution of WhiteboardVCR was to make possible the synchronization of whiteboard annotations with an audio/video streaming application such as Realplayer and Real Producer. Presenters now have the choice to pre-record the lecture before delivery, or record the lecture live during the presentation. The synchronization functions required a communication interface between external Windows applications and our Java authoring tool. We solved this problem by writing a simple Visual Basic program to control external applications and communicate with WhiteboardVCR using a TCP socket.

Mixing Human and TTS Synthetic Voice in Presentations

So far, we have discussed about the technical and linguistic merits of using synthesized speech in the presentation. We have also argued that the combined use of human speech and speech synthesis can help making the presentation much more interesting. To put our hypothesis to the test, we conducted a study for the combined speech effect on presentations. The results are discussed later.

Here are some situations that could benefit from the mixed mode:

- **Narrator-style presentation**: Like a television show, we could use TTS voice to animate different parts of the presentation or as special effects to catch the audience’s attention. For example, we could use TTS voice as a prelude to every section of the presentation, or vice versa. The contrast derived from switching from one type of voice to another could also help to emphasize important points of the presentation.
Interactive presentation: It would be possible to generate interactive event-driven course materials that ask questions to the audience. For instance, we could personalize the on-line materials by prompting the audience to type in their name, then calling their name while they are viewing the presentation. Using the Text-to-Speech style, we could produce interactive questions more easily than through audio recording, since text can be easier to create and maintain. Moreover, we could gain some degree of automation using the programmatic interface of TTS. The main portions of the presentation could be delivered by human voice as usual.

Foreign language teaching: Consider the case of English teaching in Asian countries, where not all teachers are fluent in English. Using our tool, they can use speech synthesis to read the English sentences in the lesson to the class, and then explain all or part of the lesson in their own language. This should enhance language-learning experiences for children in non-English speaking countries.

Using WhiteboardVCR

First, users need to prepare the background slide images, which they could normally get from saving their PowerPoint presentation as images or using a macro that we have written to automate this conversion step. Then, they can load the slides into the authoring tool, and start the recording process to add annotations or narrations. To start recording, users simply need to press the record button in the control toolbar. At that time the Timing Recorder Clock will start counting the time. When users add annotations (e.g., text highlighting and freehand drawing), the Record Engine will register type, location, and timing of annotations.

WhiteboardVCR works in three different modes: pre-recording, live recording, and editing. In the pre-recording mode, users can create the lecture at their leisure time and publish it to the content server for on demand viewing. In the live recording mode, the entire presentation is captured for later use, along with all the annotations and actions intact. Finally, in the editing mode, users can begin playing back a presentation until the point they want to change, then pause and add more annotations, change, or remove erroneous elements of the presentation.

In the pre-recording mode, users can add narration by pressing the pause button to halt the Clock. This brings up the TTS Manager Dialog panel. After typing in the text for the TTS agent to speak, users can press the Record button to resume the recording. Upon resuming, the speech will be spoken, so users can know how long the speech will take, and can add annotations accordingly. In the live recording mode, we cannot pause the presentation, thus users need to prepare the written speech in advance and select them from drop-down box at presentation time.

In the editing mode, users normally play a presentation until the point they want to change and then they pause and add more annotations, change, or remove erroneous elements of the presentation.

Here are some potential usage scenarios for the editor:

- To correct a live presentation made by the presenters themselves before it is published to the Web.
- To add subtitles for existing video streams. With the aid of Speech-to-Text recognition technology or a prepared script, users can add subtitles by simply adding Text-to-Speech objects for the particular video segments and set those objects to be mute.
- To emphasize important parts of a presentation. Users can choose to help changing and/or concentrate the audience’s focus on select points of the presentation, or add or replace audio at any point of the presentation with synthesized voice or record another audio stream to replace that part, if they feel using TTS voice in that part can degrade the overall quality of the presentation. Note that the default text output from the Text-to-Speech objects can be optionally disabled.

After we published a presentation using the WhiteboardVCR authoring tool, we can view the presentation by using the playback applet. The requirements for playback include a Java-enabled Web browser and Microsoft Agent (in the case that the presentation has Text-to-Speech objects). Users can pause the playback at any point, replay, or adjust the speed of Text-to-Speech objects for better understanding.
Media Objects and Synchronization Model

In this section, we describe the technical concepts behind WhiteboardVCR. The main function of the WhiteboardVCR authoring tool is to let users arrange the multimedia objects to show at their desired order and publish the presentation for playback. The media objects that WhiteboardVCR supports are:

- **Annotation Objects**: they are markups on slides, e.g., text, lines, and scribble
- **Slide-Changing Objects**: they ensure that the correct slide is displayed at the appropriate time
- **Delay Objects**: they define the playback engine waiting time before another object is considered for execution
- **TTS Objects**: they are text commands used to control the behavior of the TTS agent
- **Streaming Objects**: they are audio/video sound clips

If we consider the length of media objects, we can separate media objects in two main categories: time-bound and non-time bound. Time-bound objects always take the same amount of time to complete, regardless of system conditions. Delay objects and annotation objects are examples of time-bound objects. On the other hand, non-time-bound objects do not finish execution at a predictable time, as it may vary through network and CPU usage conditions.

For synchronization, we have two models for ordering media objects. The first model simply orders the media objects based on the start time of each object. Thus, the playback engine simply starts each object at the specified time. This is illustrated in Figure 1. The strength of this model is simplicity. It guarantees that each media object starts at the correct time and this model is widely used in Web lecture systems. It is easy to verify that this model can handle media synchronization correctly for every time-bound object. However, non-time-bound objects could cause problems if they took too long to complete. In that case, they would interfere with the activation of the next set of objects. One way to address this problem is to make such objects time bound by introducing an input field for users to specify the lifespan limit of these objects. The object will be terminated if its lifespan exceeds the specified time limit.

Since non-time bound objects are hard to fit in the timestamp model, we constructed another model to ensure the proper sequencing of events in the presentation. The model can be represented as a single-rooted directed acyclic graph of media objects as shown in Figure 2. Nodes in the graph represent media objects and the set of directed arcs defines the order of presentation of the media objects. A node will be executed after all parent nodes finished, and every node in the graph except the root must have at least one parent.

The circle-head point dashed line denotes the endpoint synchronization relationship, which forces the media object connected to the circle-head to end upon the completion of the media object on the opposite end of the link. It is easy to verify that the simple yet powerful parent-child relationship described above ensures that media objects are displayed in the correct order. From figure 2, consider node D a drawing object on a slide, and node C a narration for node D. The narration starts when the drawing starts, and the next object, node E waits for both C and D to finish before it starts.
However, in the presence of media objects whose duration depends on system resources (or simply non-time-bound objects) can cause the presentation duration longer than desired. In that case, if users want to fix the duration of a media object they can insert a delay object to terminate it at the end of the lifespan of the delay object. For example, consider node B is a slide-loading object that has a variable loading time subject to the network conditions and we want to limit it to two minutes. We employ node X as a two-minute delay object and link it to node B using an endpoint synchronization relationship. This way, the lifespan of node B cannot exceed two minutes.

The Authoring Tool

We implemented our authoring tool using Java and Visual Basic. Figure 3 shows a screenshot of the authoring tool. When the user presses any of the control buttons (i.e., play, paused, stop, and record) on the whiteboard interface, the program also sends commands to the audio/video application through ActiveX controls. The authoring tool translates all user actions into an appropriate synchronization model. The information recorded in the authoring process is sent to the content server. At that time, users can use our playback client to view the lecture over the Internet. Figure 4 shows the structure of the authoring tool.
The record engine records user slide annotations, by insert appropriate delay objects, or record the time stamp depend on which synchronization model being used. The slide loader layer let users change background images. The agent editor handles the text-to-speech object controls, and users can add voice narration through this interface.

In addition to the record engine functions, the editor has the ability to preview the presentation. Users can replay their material, pause it at any point, and modify existing objects or add new objects as they see fit.

The Playback Client

The playback client receives data from the content server. It has two view modes corresponding to the synchronization models: based on time-stamp and based on parent-child relationship. In the first mode, the playback client uses timing information from Realplayer for synchronization. It will simply start showing the objects at the appropriate time.

For the second mode of play, we perform a breadth first search travers al on the synchronization graph, using the following algorithm illustrated in figure 5.

**Algorithm:**

- **Initialization Phase:**
  - Mark the root node as Queued, its parent as Finished.

- **Execute Phase:**
  - For each node that has status Queued,
    - If all parents of this node are Finished then
      - Start playing this object.
      - Mark all children of this node as Queued.
      - Mark this node as Played.
  - Repeat until Queued node does not exist.

First, we define states of media objects in the algorithm as follows: **Idle:** object is idle, **Queued:** object starts waiting for its parent to finish, before start its execution, **Played:** object starts playing; **Finished:** object finished playing. We start by marking the root node as Queued, and we repeatedly check for the Queued nodes whether they satisfy the condition that all parents are finished. If satisfied node is found, we start its execution, and change its children status to Queued. We repeat the process until the queue is empty.

When an object has finished playing, we check whether the endpoint synchronization field exists. If it does, we terminate the specified object. We use this termination procedure to control the scheduling of non-time bound objects (Figure 6).
Evaluation

We compared the merits of using TTS voice versus human voice in our presentation tool. We also studied the benefits and effects of mixing TTS voice and human voice together to make a presentation more interesting.

We performed a survey by asking twenty-four people to participate as the audiences: of which twelve people were native English speakers and twelve were non-native. In the experiment, we had a total of twelve distinct presentations, six by native English speakers and six by non-native English speakers. Non-native participants had their most recent TOEFL scores in the region between 500 and 600. We employed three narration styles for each presentation: original human voice, Text-to-Speech synthesis, and mixed. We did not let anybody attend a same presentation more than once, since the participants might not be able to judge impartially about the degree of understandability of a presentation if they had viewed the same presentation twice. We assigned a set of six presentations of different subjects to each participant to watch. Each presentation in the set had a different combination of language proficiency of the presenter and style of narration (e.g., Native Human, Native TTS, etc). At the end of each presentation, we asked the audiences to answer five quiz questions and give the rating of the overall impression of that presentation using a scale of one (lowest score) to ten (highest score). The results of the evaluation are shown in Table 1 and 2.

<table>
<thead>
<tr>
<th>Presenters</th>
<th>Native</th>
<th>Non-Native</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Median</td>
</tr>
<tr>
<td>Native Human</td>
<td>4.42</td>
<td>5</td>
</tr>
<tr>
<td>Native TTS</td>
<td>4.58</td>
<td>5</td>
</tr>
<tr>
<td>Native Mixed</td>
<td>4.58</td>
<td>5</td>
</tr>
<tr>
<td>Non-native Human</td>
<td>2.42</td>
<td>3</td>
</tr>
<tr>
<td>Non-native TTS</td>
<td>4.17</td>
<td>4</td>
</tr>
<tr>
<td>Non-Native Mixed</td>
<td>3.83</td>
<td>4</td>
</tr>
</tbody>
</table>

*Table 1. Correctness results for the different narration styles and level of language proficiency*

<table>
<thead>
<tr>
<th>Presenters</th>
<th>Native</th>
<th>Non-native</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Median</td>
</tr>
<tr>
<td>Human Voice</td>
<td>7.33</td>
<td>7</td>
</tr>
<tr>
<td>TTS</td>
<td>7.13</td>
<td>7</td>
</tr>
<tr>
<td>Combined</td>
<td>7.58</td>
<td>7.5</td>
</tr>
</tbody>
</table>

*Table 2. Presentation impression scores*

Because the sample size in our survey is too small, we analyze the collected experimental data using classic nonparametric methods. We have data in the form of number of correct quiz answers and impression scores for each person. We use Mann-Whitney Test in the case of two sample sets and Kruskal-Wallis Test in the case of three sample sets. We use the generally agreed level of statistical significance. If the P value is smaller than 0.05 then the result is significant (i.e. the sample sets are different). Table 3 shows result from nonparametric analysis.

Our experimental results clearly show that in general, the native audiences, due to their innate command of the language, could understand the presentations in all forms better than their non-native counterparts could. In contrast, there is evidence that the non-natives might have some difficulties presenting in English, as suggested by the lowest score of understanding for both the native and non-native audiences. When their voice was substituted entirely by TTS voice, understandability improved very significantly, as suggested by the average
score jumped from 2.17 to 4.04 and significance estimated using the Mann-Whitney statistic (Z= 4.57, P=0.0000).

However, for the native presenters, using TTS voice did not significantly change the understandability for both the native and non-native audiences, as suggested by the Mann-Whitney statistic (Z= -1.54, P = 0.0623), although the average score rose from 3.75 to 4.25. Since the native presenters normally speak more clearly than the non-natives do, the use of TTS contributes less significantly to the understandability enhancement. If we consider the improvement only on the non-native audiences, there is enough evidence to conclude that TTS did enhance understandability (Z = 1.86, P = 0.0316) although at a weaker level comparing to that of non-native presenters. We believe that the enhancement is due to the speaking rate of the TTS voice, which was slower than that of human speech, non-native audiences were able to follow the presentation more easily.

<table>
<thead>
<tr>
<th>Compared Groups</th>
<th>Human</th>
<th>TTS</th>
<th>Mixed</th>
<th>Analysis Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNP, between TTS and Human</td>
<td>Mean = 2.17</td>
<td>Mean = 4.04</td>
<td></td>
<td>Z = 4.57</td>
</tr>
<tr>
<td></td>
<td>Median = 2</td>
<td>Median = 4</td>
<td></td>
<td>P = 0.0000</td>
</tr>
<tr>
<td>NP, between TTS and Human</td>
<td>Mean = 3.75</td>
<td>Mean = 4.25</td>
<td>Mean = 4.5</td>
<td>Z = -1.54</td>
</tr>
<tr>
<td></td>
<td>Median = 4</td>
<td>Median = 4.5</td>
<td></td>
<td>P = 0.0623</td>
</tr>
<tr>
<td>NP and NNA, between TTS and Human</td>
<td>Mean = 3.08</td>
<td>Mean = 3.92</td>
<td>Mean = 4</td>
<td>Z = 1.86</td>
</tr>
<tr>
<td></td>
<td>Median = 3</td>
<td>Median = 4</td>
<td></td>
<td>P = 0.0316</td>
</tr>
<tr>
<td>NA, between Human, TTS and Mixed</td>
<td>Mean = 3.42</td>
<td>Mean = 4.38</td>
<td>Mean = 4.21</td>
<td>H = 5.76</td>
</tr>
<tr>
<td></td>
<td>Median = 4</td>
<td>Median = 5</td>
<td>Median = 4</td>
<td>P = 0.0561</td>
</tr>
<tr>
<td>NNA, between Human, TTS, and Mixed</td>
<td>Mean = 2.50</td>
<td>Mean = 3.92</td>
<td>Mean = 3.42</td>
<td>H = 15.20</td>
</tr>
<tr>
<td></td>
<td>Median = 3</td>
<td>Median = 4</td>
<td>Median = 3.5</td>
<td>P = 0.0005</td>
</tr>
<tr>
<td>NNA, between Human and Mixed</td>
<td>Mean = 2.50</td>
<td>Mean = 3.92</td>
<td>Mean = 3.42</td>
<td>Z = -2.50</td>
</tr>
<tr>
<td></td>
<td>Median = 3</td>
<td>Median = 4</td>
<td>Median = 3.5</td>
<td>P = 0.0062</td>
</tr>
<tr>
<td>NNA, between TTS and Mixed</td>
<td>Mean = 3.92</td>
<td>Mean = 3.42</td>
<td>Mean = 3.42</td>
<td>Z = 1.47</td>
</tr>
<tr>
<td></td>
<td>Median = 4</td>
<td>Median = 3.5</td>
<td>Median = 3.5</td>
<td>P = 0.0704</td>
</tr>
<tr>
<td>Impression, between Human, TTS, and Mixed</td>
<td>Mean = 7.27</td>
<td>Mean = 7.33</td>
<td>Mean = 7.48</td>
<td>H = 1.62</td>
</tr>
<tr>
<td></td>
<td>Median = 7</td>
<td>Median = 7</td>
<td>Median = 7.5</td>
<td>P = 0.4441</td>
</tr>
</tbody>
</table>

Table 3. Nonparametric analysis results
Note: NA = Native audiences, NNA = Non-native audiences, NP = Native Presenters, NNP = Non-native presenters

Next, we analyze the understandability using the mixed style, which is mainly human voice with TTS substitution in some parts. For the native audiences, we cannot single out any style as the best or worst, since the computed Kruskal-Wallis statistic H is 5.76 (P = 0.0561). However, the same computation for the non-native audiences yields H 15.20 (P = 0.0005). This suggests that those three styles are significantly different for the non-native audiences. We further analyze the difference of non-native audiences scores between human voice and mixed, and between TTS and mixed. The results (P = 0.0062 and P=0.0704, respectively) show that for non-native audiences, the mixed style improved understandability with respect to the human voice style and did not perform significantly worse than the pure TTS style.

Next, we analyze the survey of audiences’ impressions. The result from Kruskal-Wallis test shows that there is no clear winner here (H = 1.62, P = 0.44) among the three styles of narration. This suggests that each style has its own merits and the style selection is subject mainly to the user’s preference. However, from looking at the average scores, we note that there are some small variations between the native and non-native English speakers, although not statistically significant. We found that the native English speakers had the tendency to prefer the mixed mode to other styles, provided that the TTS voice is properly placed. On the other hand, non-native English speakers still opted for the easier-to-understand version among all the styles. Some native audiences reported that the speaking rate was too slow for them. Furthermore, they felt that the flat intonation of the TTS voice was not conducive to any emotional affinity between the presenter or presentation and the audience.
Discussion

One feature of the presentation tool is synchronization of slide annotations with another media. That media can be streaming audio and/or video as well as synthesized audio. A few research and commercial systems support synchronized slide annotations with audio and video. Common to all the systems is the requirement to capture the annotations with timing information. From the standpoint of playback strategy, they differ in terms of whether the entire content is delivered (streamed or downloaded) as a single media file or split into parts (audio and video in one media file and annotated slides in separate HTML documents) before sending.

The former method has two major drawbacks. Different data types are typically stored in different tracks of the media file, but the available number of tracks is fixed. Hence, only a limited number of annotation effects can be included in the media file. Another problem is that unless the media file is streamed, users need to download the entire file even if they only want to view a part of the presentation. A recent development using this approach is Severence (2000).

The split approach, which is the strategy we adopted, does not have any of the above limitations. The playback is based on separate players for audio/video and slides that are synchronized with each other by accessing the API of the audio/video player. Work in this area is exemplified by Klevan & Vouk (1999) and Classroom 2000 (Gregory, et al., 1998). In WLS, every slide is a HTML document and the annotations are entered in a Java applet that works like a whiteboard. All media can be replayed in the same sequence as in the original presentation using a trigger from the audio player. In Classroom 2000, every slide and in-slide annotations are stored in a HTML document. Slide markings captured from a hardware whiteboard are converted automatically into static image maps.

Using Realpresenter users can use text as another media in synchronization. Oratrix’s GRiNS (Oratrix, 2002) is an authoring tool based on SMIL (Synchronized Multimedia Integration Language), which allows users to create presentations using a variety of media types. Both Realpresenter and GRiNS can deliver presentations for different connection speeds. In Eloquent (Eloquent, Inc., 2001) the transcript of the narration is scrolled in synchronization with the narration and slides. Audiograph (NZEdSoft, 1999) is an example of another tool to create an online multimedia Web presentation. Nevertheless, we are not aware of any existing work that applies TTS technology in the creation of synchronized slide presentations for the Web. Moreover, when TTS technology is enhanced with machine translation capabilities, the same presentation can be delivered in other languages.

Conclusion

We have presented WhiteboardVCR, a new Web lecture production tool utilizing Text-to-Speech technology. WhiteboardVCR provides user with the ability to choose their delivery style in their presentation, live or pre-recorded. Like any other presentation tools, the presenter can prepare a quality lecture once and will be able to reuse it many times and refine it as needed. In addition, the cost of production and maintenance of Web lectures in our approach is lower as audio editing is more difficult than TTS editing. In the case that the entire Web lecture is delivered using TTS, both storage and bandwidth requirements are significantly lower.

The study has both validated our intuitions about the benefits of TTS technology in Web lectures, and revealed the strengths and weaknesses of the different narration styles. Although TTS technology still cannot model fully the speaker and speaking style effects, it has already proven quite useful in a number of contexts. This work demonstrates the merits of TTS in Web lectures, in the pure mode where the entire lecture is narrated using TTS or in the mixed mode where the lecture contains both speech synthesis and human voice. In the pure mode, we have found that it is especially suitable for audiences with language understanding problems, whereas in the mixed mode, it is very useful for making the lecture more interesting, in scenarios such as: online interactive Web lectures and foreign language education (without the need to hire a native speaker).

In the future, we will focus in the following directions: stabilize the research prototype, continue to adapt our system to the latest TTS technology, make the authoring tool more user-friendly, evaluate the ease of use, and develop more interactive components (e.g., hyperlinks, animation effects) that users can include in their presentations.
Acknowledgement

We are grateful to the reviewers and our research colleagues at Media and Technology Laboratory/United Nations University for helpful comments and suggestions.

References


W3C SYMM Working Group (1998), Synchronized Multimedia Integration Language (SMIL), http://www.w3.org/TR/REC-smil/