LiteMinutes: An Internet-Based System for Multimedia Meeting Minutes

Patrick Chiu, John Boreczky, Andreas Girgensohn, Don Kimber

FX Palo Alto Laboratory
3400 Hillview Avenue, Bldg. 4, Palo Alto CA 94304, USA
http://www.fxpal.xerox.com/

{chiu, johnb, andreasg, kimber}@pal.xerox.com

ABSTRACT

The Internet provides a highly suitable infrastructure for sharing multimedia meeting records, especially as multimedia technologies become more lightweight and workers more mobile. LiteMinutes is a system that uses both the Web and email for creating, revising, distributing, and accessing multimedia information captured in a meeting. Supported media include text notes taken on wireless laptops, slide images captured from presentations, and video recorded by cameras in the room. At the end of a meeting, text notes are sent by the note taking applet to the server, which formats them in HTML with links from each note item to the captured slide images and video recording. Smart link generation is achieved by capturing contextual metadata such as the on/off state of the media equipment and the room location of the laptop, and inferring whether it makes sense to supply media links to a particular note item. Note takers can easily revise meeting minutes after a meeting by modifying the email message sent to them and mailing it back to the server's email address. We explore design issues concerning preferences for email and Web access of meeting minutes, as well as the different timeframes for access. We also describe the integration with a comic book style video summary and visualization system with text captions for browsing the video recording of a meeting.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – *architecture, navigation, user issues.*

General Terms

Documentation, Design, Human Factors.

Keywords

Meeting support systems, meeting capture, note taking, hypermedia systems, video applications, multimedia applications.

1. INTRODUCTION

Documenting meetings can be an important part of organizational activities. Meeting minutes constitute a portion of the organizational memory. Right after a meeting, it is often useful to look at the notes to review and act on decisions. Even during a meeting, it can be helpful to refer to something from a point earlier in the meeting, for example, asking a question that pertains to an earlier presentation slide.

Multimedia meeting minutes provide a rich record of what took place in a meeting. Video picks up details that are difficult to catch, captures nonverbal activity, and shows context. Slides contain text, images and meaningful layout information. Meeting minutes, when correlated and linked to the video recording and slides, can be used to retrieve and playback interesting points of a meeting.

As multimedia applications become more lightweight and the quality of multimedia over networks improves, supporting the creation and access of multimedia meeting minutes over the Internet becomes more compelling. Also, workers are becoming more mobile and distributed, and the Web and email are now indispensable for collaborative activities. In view of these two trends, we have designed and built LiteMinutes, a system for multimedia meeting minutes that uses both the Web and email for creating, revising, distributing, and accessing multimedia information captured in a meeting.

The Web provides a set of technologies that allows users to access meeting minutes and other information about a meeting without having to install special-purpose applications. The Web offers an infrastructure facilitating both access by remote users and communication among the different components of our system. Our goal of both lightweight meeting minutes capture and lightweight access is well-supported by this infrastructure.

There are several challenges to designing a system that works well. The first is how to record the video and capture the slide images; for this we have equipped a conference room at our laboratory for multimedia meeting capture [5]. Our approach is that a multimedia meeting room serves as a computing environment that captures the heavier weight media such as video, audio, and slide images. These are then combined with the more lightweight and interactive medium of notes taken during a meeting.

The second challenge is designing an Internet-based application for taking notes in a meeting. From our experience with

multimedia note taking systems (see [4], [6], [23]), we found that a note taking application must support rapid interaction. Taking notes during a live event required users to pay close attention and sometimes they had to participate in the meeting in addition to formulating notes. This made it difficult for novice users to fiddle with user interface widgets and perform tasks such as labeling or organizing information. After some experiments with various hardware devices and software using HTML forms and applets, we arrived at a minimalist Java Swing applet for taking text notes on wireless laptops.

The third challenge is generating hypermedia meeting minutes from the text notes and the recorded media. In order to do this, the text notes must be timestamped and parsed into note items, and the individual note items linked to each media. We discovered that people could be confused when links are generated indiscriminately; for example, supplying a link to the slides when no slide was used during that part of the meeting. To deal with this, we developed a technique for smart link generation using contextual metadata.

Finally, the fourth challenge is distributing and revising the hypermedia meeting minutes generated by the system. We studied the work process of the note takers and found that it was best to use email to distribute and revise the notes. We also surveyed the recipients who indicated both email and Web versions of the meeting minutes should be available for access. To support revisions on the Web, the system automatically detects the email revisions and updates the meeting minutes collection on the Web.

This paper is organized as follows. We discuss our empirical observations and design requirements in section 2. Next, we describe the LiteMinutes application in section 3 and the system architecture in section 4. In section 5, we describe a video summary and visualization system with text captions for browsing a video of a meeting on the Web. To wrap up, we discuss related work in section 6 and conclude with section 7.

2. EMPIRICAL OBSERVATIONS AND DESIGN REQUIREMENTS

To inform the design of our meeting minutes system, we observed the pre-existing practice of creating and distributing meeting minutes, conducted a survey on user access preferences, and explored various devices and user interfaces.

2.1 Pre-Existing Practice

Before introducing the first LiteMinutes prototype, text-based email meeting minutes had been in use for more than a year and a half at our lab. Notes were taken on paper by a single person, transcribed and sent out as an email to all lab members. Figure 4a shows a sample of these notes. The transcription was typed using an email application and not a word processor. Transcription was a tedious process, and the note taker often had to track down people in the meeting to clarify what was said or to obtain information that was shown on a slide. Part of this was due to the difficulty of catching everything in a meeting. Another factor was that the spelling of names in cosmopolitan Silicon Valley and the spelling of technical terminology sometimes needed checking.

Four people took notes over this period of a year and a half. For a span of several months, one person served as the primary note taker with the other three occasionally substituting. Then another person rotated in as the primary note taker. These four were part of the staff at the lab but not researchers, and were inexperienced users of computer technology.

Our observations suggest that the device and application for taking meeting minutes must be easy to use and require at most a few minutes of training or re-familiarization when someone has to substitute for the primary scribe. Because the notes eventually end up in email form, providing integration with email would be useful.

2.2 Survey on Access Preferences

On the access end, we conducted a survey to determine the preferences for accessing meeting minutes. A key question was whether people like to have meeting minutes delivered as email or to have the minutes on the Web for browsing; i.e. the push/pull question.

In our survey, we walked around the building and interviewed 13 people in their offices. We asked them two questions:

1) Do you read the email meeting minutes that you have been receiving?

2) Would you prefer to have the minutes: (a) emailed to you, (b) put on the Web, or (c) both?

They were also given an opportunity to make comments after answering these questions.

For question (1), 11 of the 13 subjects answered that they read the emails. One subject commented that he "looked more carefully if missed the meeting." Another read them for the "spin."

For question (2), 5 preferred email, 2 preferred the Web, 5 preferred both, and 1 said it "doesn't matter." One subject commented: "won't go to the Web to look, only email." Another commented that it was "easier to find things on the Web than through email." Yet a third person saved all the email minutes and felt that he could find things by searching through them.

This survey indicates that in order to support the habits of users of email and the Web, it is desirable to have both email delivery and Web access to the meeting minutes.

2.3 Exploring Devices and User Interfaces

We found laptop computers with a wireless network connection to be well suited for taking meeting minutes. Laptops are familiar devices that require no training for someone already familiar with a PC workstation and its keyboard interface. Hooking it up to a wireless network allows the note taker to sit anywhere in the room and provides an unobtrusive form factor. While text is limited in expressiveness compared to ink (which supports both writing and drawing), having the text entered while note taking during a meeting saves the scribe from the time consuming task of transcribing the notes after a meeting.

We also explored other devices such as pen-based notebook computers, scanned notes on paper handouts, and hybrid paper/digitizer CrossPads [7]. Pen-based computers have not yet reached the point where writing on them feels like writing on paper, plus there are problems with resolution and parallax that take time to get used to. Multimedia pen-based note taking systems such as FXPAL NoteLook [6] can create notes with slide images, pictures of the room activity, along with ink notes and

annotations, but such full-featured systems require more than a few minutes of training. Scanning notes on paper can be useful in certain situations such as when handouts are available (see [4]), but it is generally a difficult problem to determine time information of the ink strokes written with ordinary pens. Hybrid paper/digitizer systems such as the Audio Notebook [19] and the CrossPad allow a user to write on a paper pad on top of a digitizer that captures the ink strokes electronically while timestamping them. These systems are quite usable, but have some quirks in the user interface such as provisions for letting the system know what page the writer is on.

When notes are shared publicly, legibility is crucial. The NotePals [9] work shows that people prefer reading text notes to handwritten notes taken on PDAs. Current handwriting recognition technology does not solve the transcription problem, due to insufficient accuracy when applied to transcribing handwritten meeting notes.

2.4 HTML Form-Based Prototype

Our first prototype was a HTML form-based application (see Figure 1). It supports note taking, editing, and playing back of the video recording. Each note item is typed into a form field, and pressing the Enter button submits the item, which is appended to the list of note items on the page. On the left margin next to each note item are buttons for Edit and Play. Pressing the Edit link of an item puts the text of that item into the form field for editing and re-submitting. Pressing the Play link of an item plays back the video at the time that item was entered. Playback and management of the video recordings are handled by the FXPAL Metadata Media Player and MBase system [11], which are described in more detail in a later section.
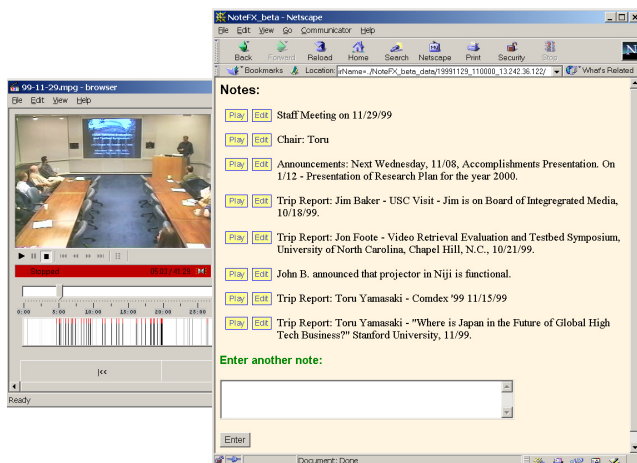

Figure 1. An early HTML form-based prototype for taking, editing, and playing back multimedia meeting minutes.

We learned a number of things from using this prototype for four months. Notes were taken in five staff meetings and a few other meetings. Forms are simpler than applets, but the interaction and user interface layout is far from ideal. Having to enter each note item proved to be tedious, and being able to specify precisely the text and time of each item was not enough of a benefit. We concluded that it was better to go with an applet with a text window for entering notes. Each character would be timestamped by the applet, and line breaks would be used later to parse the text

into separate note items for linking to media. Another observation was that note items can be longer than one line (see Figure 1). This meant that support for word-wrap was required; otherwise line breaks in the middle of sentences would appear, put in by users as a way to keep text visible in the window. The need for word-wrap made us choose Java Swing text widgets, and not Java AWT widgets that lack support for the word-wrap feature. Of course, the superior look-and-feel of the Swing user interface components was a plus.

For revising the notes, editing almost always took place right after a meeting to supply missing words or to correct typos. After this revision, further editing was never done in these samples. The text notes did not need to be perfect, because they were linked to a video recording that provides an accurate account of what took place in the meeting. Consequently, always having the Edit buttons on the page was not a good use of screen real estate. As we would find out later, people felt that links to different media (e.g. video, slides) should not exist when the corresponding media was not recorded at all or not used at a time a particular note item was taken. To solve this problem, we developed a scheme for smart link generation using contextual metadata.

2.5 Design Requirements

To summarize, we list the design requirements obtained from our observations, survey, and working with various devices and prototypes.

- Provide both email delivery and Web access
- Wireless laptop for taking meeting minutes
- Java Swing applet for text note taking
- Timestamp note items for correlating with media
- Support revision right after a meeting
- Smart link generation using contextual metadata

3. THE LITEMINUTES APPLICATION

With knowledge gained from the aforementioned observations and prototype testing, we designed and built a system called *LiteMinutes*. It has been deployed and used for our weekly staff meeting minutes for more than 10 months. In this section, we describe the typical scenario of how the application is used to create, revise, distribute, and access multimedia meeting minutes. How the system works internally is explained in section 4 below.

3.1 Creating Meeting Minutes

Creating multimedia meeting minutes with LiteMinutes is very easy: meeting participants or a designated scribe walk into a meeting with their laptops or use the wireless laptops supplied in the room (see Figure 2). The system can support more than one note taker simultaneously; sets of notes taken on different laptops are handled separately. Normally, a single scribe takes notes in our staff meetings. On a Web page that can be found from our laboratory's (internal) home page, the user clicks on a link to get the note taking applet. A screen shot of this applet is shown in Figure 3. The user simply takes notes in a text window. In the mean time, the meeting is recorded on video and slide images shown by the speaker are captured. At the end of the meeting, the user enters his email address and presses the Create Notes button. The notes are sent to the notes server via a CGI script for processing and distribution.

Figure 2. Meeting room with large main display, secondary display, and wireless laptop for note taking.
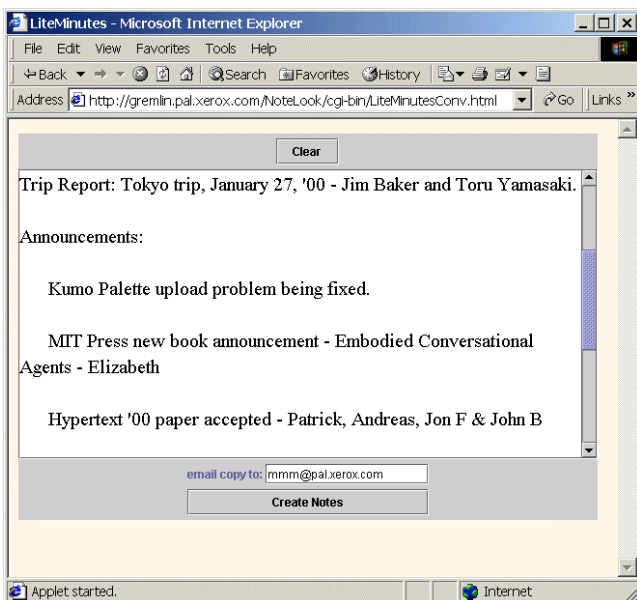

Figure 3. LiteMinutes note taking Java Swing applet running inside a Web browser on a laptop.

The notes are parsed by the server and formatted in HTML, with a small addition at the end of each note item, where a [video] and/or [slide] link is appended. Figures 4b and 4c illustrate two design alternatives for showing the links, the first with text and the second with thumbnail images. In keeping with the spirit of making a lightweight application, we chose the minimalist design with text links. Another consideration is that sending a bunch of thumbnail images in an email message either takes up space or requires HTTP access to the image host when viewing the message.

Parsing the text notes from the applet is based on the rule that newline characters separate note items, and an item is associated with the time when the first character of that item was typed. The multimedia note items are context-aware: an item's link to a medium is generated only if that medium was recorded or used at

the time that item was taken during the meeting. This is determined by checking the contextual metadata collected during the meeting.

```
Trip Report: American Productivity  & Quality Center (APQC) – Houston,
Texas ,  February 7, '00  – Steve Smoliar.

Announcement:  Masao announced that  Mr. Kato from NTT will be visiting
Tuesday, 2/15.
```

Figure 4a. Sample of early text-only email meeting minutes.

Figure 4b. Sample of HTML formatted multimedia meeting minutes in LiteMinutes. Links are generated only if that medium (slide or video) was used or recorded at the time the note item was taken.

Figure 4c. Alternative to 4b with thumbnail links, which we felt was too cumbersome for email.

After parsing and generating the HTML hypermedia meeting minutes, the server emails them to the email address filled in by note taker and a copy is placed in the Web collection. Using email for distributing the notes leverages an existing document routing system and integrates smoothly with the pre-existing work process. Since the hypermedia meeting minutes are very similar to the pre-existing text-only email meeting minutes (compare Figures 4a and 4b), the transition experienced by the users was smooth and not disruptive.

Finally, placed at the bottom of the HTML meeting minutes are links to Revisions, Weekly Meeting Notes, and Help (see Figures 5 and 6). The Revisions link accesses the latest revision of the meeting minutes. The Weekly Meeting Notes link takes the user to a Web page where the collection of notes can be browsed by the week or month (see Figure 7).

3.2 Revising the Meeting Minutes

Revision is a crucial step in the meeting minutes process that gives an opportunity for the filling in of missed items, checking details, and fixing typos. This step is easily performed in the note taker's own email application when the note taker receives the processed HTML meeting minutes as an email message sent by the LiteMinutes system. What is required is that the email application supports HTML editing, and this is a feature in popular email applications such as Microsoft Outlook and Netscape. After editing this email message, the revision is then forwarded by email to all the meeting participants and other interested parties, along with the email address for the LiteMinutes server. When the LiteMinutes system receives a revision (tracked by HTML comment tags embedded in the meeting minutes explained in Section 4.5), it automatically updates the Web collection of meeting minutes.
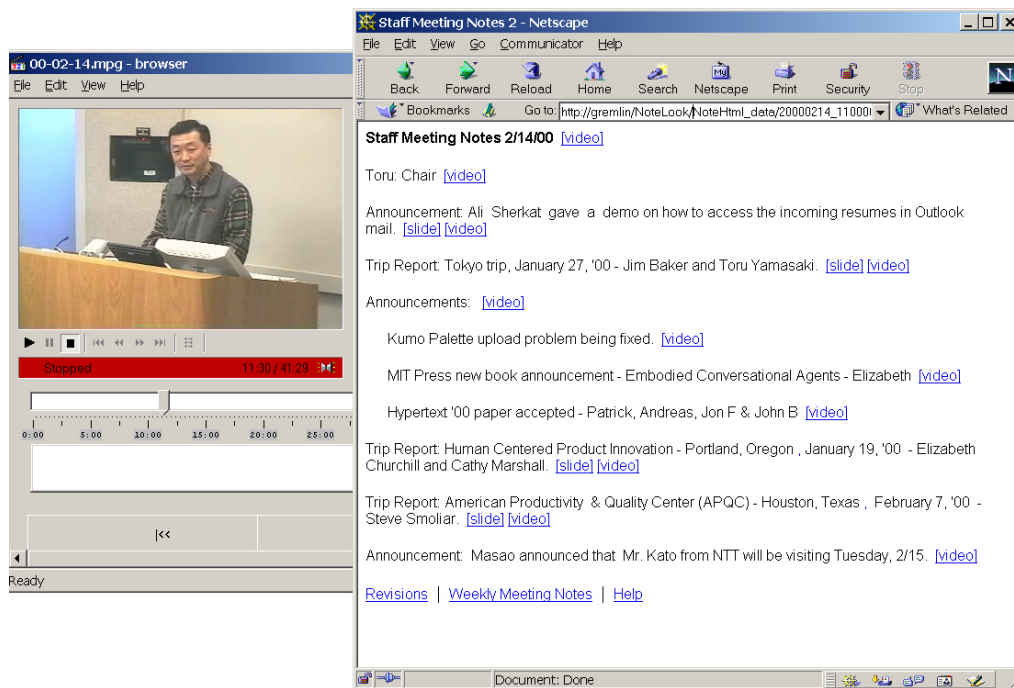
Figure 5. Video playback from meeting minutes is activated by clicking on the [video] links. Meeting minutes in a Web browser is shown on the right. The video player shown on the left is the FXPAL Metadata Media Player.
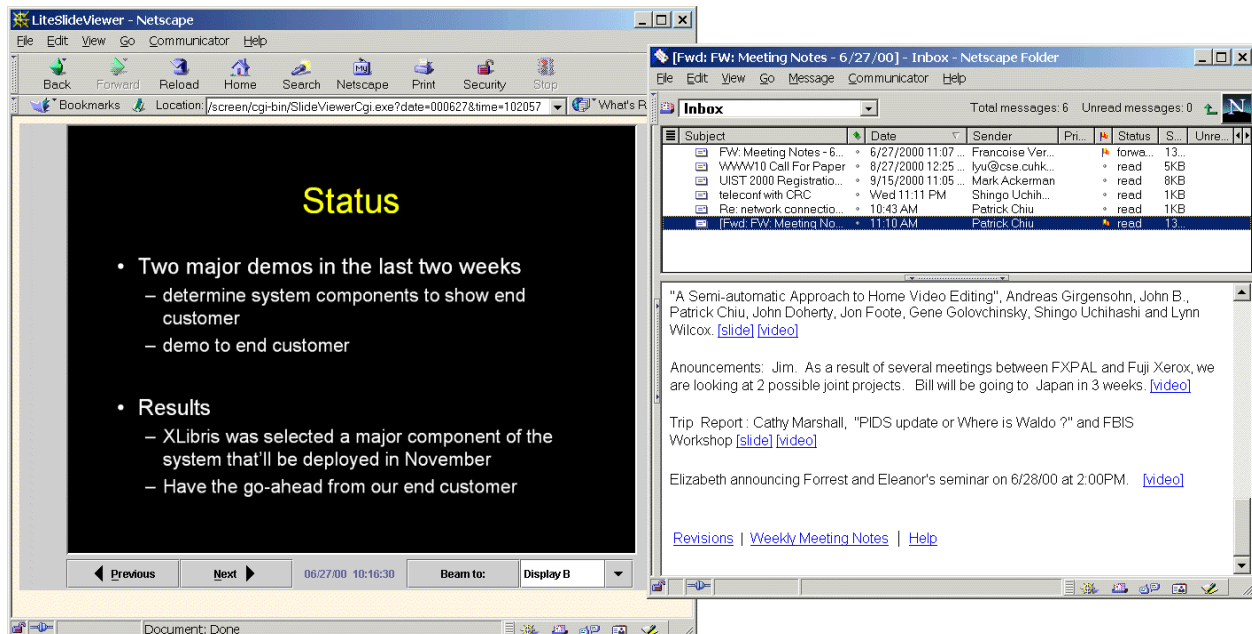

Figure 6. Slide images are accessed from the meeting minutes by clicking on the [slide] links. Meeting minutes in an email application is shown on the right. Slides are viewed in the LiteSlideViewer applet shown on the left.
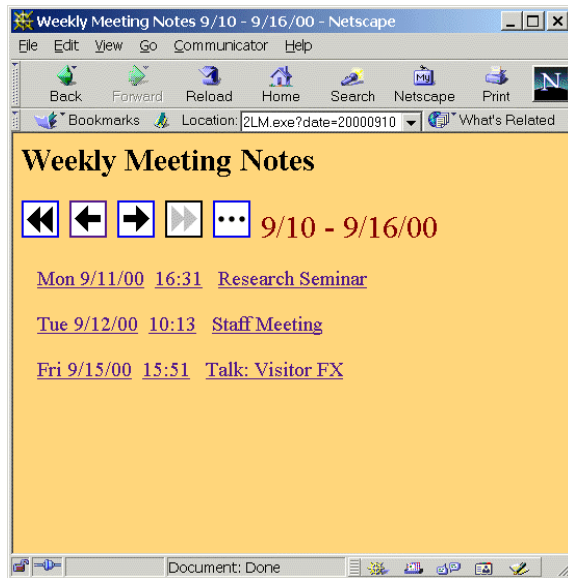
Figure 7. Browsing the archived notes on the Web. Notes are listed by the week, and users can navigate by the week or month with the arrow buttons. Real-time notes are accessed by the "…" button.

Sometimes revisions are not necessary and the note taker wants to distribute the original email meeting minutes directly. This is more appropriate for meetings that are more informal than staff meetings, such as a small group discussion on project design. In this case, the note taker simply enters several email addresses or the group's email address in the note taking applet.

An example is illustrated in Figure 6. The email message displayed is a revision.[1] The first three items of the meeting minutes shown in the email window had been revised as follows:

1) The first item, originally incomplete, was an announcement of a publication, and the details of the citation were obtained by the note taker after the meeting.

2) The second item was an announcement that the note taker had missed in the meeting. Originally the item entered was "Jim announced".

3) The third item was also partially missing in the original notes: "Trip report: Cathy, Where is…"

In the revision shown in Figure 6, we see that these incomprehensible and incomplete note items are now corrected.

The note takers who had used both LiteMinutes and the earlier HTML form-based prototype expressed that revision by editing email is highly preferred over modifying a Web page.

3.3 Accessing the Meeting Minutes

Multimedia meeting minutes can serve a number of different purposes depending on when they are accessed. Three important timeframes along with some examples that we have identified are:

---

[1] The contents of the data taken from real meetings in this example and throughout the paper have been altered for privacy reasons.

- During a meeting (real time)
  - Question and discussion referring to a slide shown earlier in the meeting
  - Instant replay of a remark or event with the recorded audio or video
- Right after a meeting (minutes to days)
  - Review details, action items
  - Skim email minutes if missed the meeting
- After an indefinite duration (weeks to months)
  - Recall plans and accomplishments over past year
  - Organizational memory

During a meeting (in real time), any meeting participant can view the live meeting minutes on the Web from their laptops while the scribe or others are taking notes. These real-time notes are accessed through the "…" link on the Weekly Meeting Notes page (see Figure 7). The real-time notes look exactly like the regular HTML-formatted notes. Clicking on the [slide] link of a note item shows that captured slide image in our LiteSlideViewer applet (shown in Figure 6). This applet can be used to navigate through the presented slides.

For purposes of question and discussion, a slide image may be "beamed" to the secondary display in our meeting room (see Figure 2). The LiteSlideViewer applet is designed to be context-aware and beaming slide images is operational only if the laptop is inside the room during the meeting, with the available displays in that room listed in the combo box at the lower right of the applet. The applet has a "Beam to:" button for sending a slide image to a selected room display. In a room with multiple displays, the default display is the secondary display because a slide shown by someone who is not the presenter should not intrude upon the main display used by the presenter.

Having instant replays in meetings is an interesting capability. Rather than having people repeat what they said, an instant replay can be invoked. The playing back of the audio or video recording during the meeting itself has been investigated by our colleagues at Xerox PARC in the WhereWereWe project [14]. Our Metadata Media Player does not currently support this capability, although we may add this capability in the future.

Right after a meeting (within minutes to days), the recipients of the email can skim the notes quickly, or browse an identical set of notes on a Web page (see Figures 5 and 6). Clicking on the [video] link of a note item brings up a video player (shown in Figure 5). Clicking on the [slide] link shows the corresponding slide image, which is displayed by our LiteSlideViewer applet on a Web page (see Figure 6).

After some weeks or months, the Web archive provides a better way to retrieve and access the meeting minutes. Currently we list sessions by the week, and users can flip back and forth by the week or by the month (see Figure 7). Providing search capability would be desirable, and we intend to add this in the future. People who file away all of their email meeting minutes can search them with their email application.

The meeting minutes are also accessible from our MBase system, which has a listing its video collection. Another use of the minutes is for text captions in our Manga visual summary of a video. Both systems are described in the next section.

4. SYSTEM ARCHITECTURE

In this section we discuss the LiteMinutes system architecture. We describe the various components of this multimedia system: video management, slide image capture, smart link generation with contextual metadata, and revision via email.

4.1 LiteMinutes Components

Each medium is handled separately by a server that manages the capture and playback of that medium (see Figure 8). Metadata about the contextual or environmental conditions are also captured. This loose coupling of the media modules provides flexibility when supporting different combinations of media for different kinds of meetings, and allows the multimedia services to be offered to other applications beyond multimedia meeting minutes. Furthermore, extending support to a new medium is simple.

The LiteMinutes applet for taking meeting minutes is a Java 2 Swing applet. When a user types a character, the applet timestamps it. This applet runs in any Web browser that supports Java 2 or has a Java 2 plug-in. Each time a line break character is entered, the text notes along with timestamps for each character are sent from the applet to the server. This continual updating enables the notes to be shared and viewed in real time, accessed from the "…" button on the Weekly Meeting Notes Web page (see Figure 7). Continual updating also acts as auto-save, which prevents all the notes from being lost in case of a breakdown on part of the applet, server, or wireless network. A CGI script on the server parses the notes and generates the multimedia meeting minutes in HTML with the video and/or slide links. These minutes are then put on the Web. If the user is performing a Create Notes operation and the email address field in the note taking applet is filled in, the server also emails a copy of the HTML meeting minutes via a SMTP mail server.
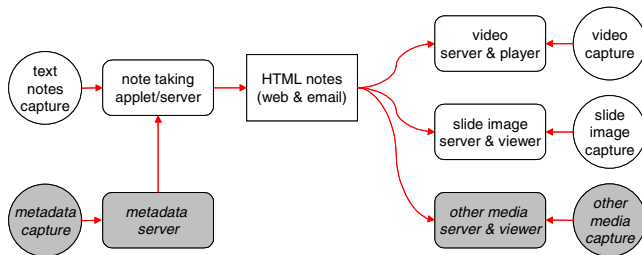

Figure 8. Diagram showing how the text notes, different media, and metadata are captured and connected.

For viewing, real-time reviewing, and beaming of slide images, an applet called *LiteSlideViewer* is used (shown in Figure 6). The [slide] link of a note item is a CGI script that fetches the LiteSlideViewer applet and shows the target slide image. The applet has buttons to navigate to previous and next slides. The timestamp of the slide is shown. For beaming a slide image to a room display, a combo box provides a choice of available displays. The beam button communicates back to the slide image server the command to show the image on the target room display.

An applet called *LiteSlideShow* displays the beamed slide images on the room display. Figure 2 shows this applet running on the secondary display in the room. The LiteSlideShow applet periodically checks for new images from the slide image server.

Using these applets also provides a simple way to share and beam images to remote displays during a teleconference.

4.2 Video Management

For the video server and player we use the FXPAL MBase system with its Metadata Media Player [11]. The video is recorded directly in MPEG and can be played back right after a meeting. The clocks of the workstation recording MPEG and the laptop for taking meeting minutes are kept synchronized with the Network Time Protocol. This protocol provides sufficient accuracy to correlate events to the right frame in the video. We also convert the MPEG video to RealVideo, which is easier to stream over the Internet. The MBase system provides the necessary support to automatically choose the video format most appropriate for the client. For local users, MPEG served via Microsoft Windows file sharing to our Metadata Media Player provides better performance than any streaming video format. For remote users, RealVideo is the appropriate choice. The MBase system uses a combination of JavaScript and a helper application for the MPEG playback to serve the video. A CGI script generates JavaScript to set a boolean variable based on the client's IP address, which determines the video format. The combination of static and dynamically generated JavaScript allows for the inclusion of JavaScript video links in the static meeting minute pages that either launch a helper application or open another browser window with a RealVideo plug-in. Unfortunately, these JavaScript links are not supported by all email applications so that MPEG playback is the only choice from the email messages. Overall, it is possible to use any video player that provides an API function to play a video at a given point in time. We note that an audio recording may be used instead of video.

The video can be shot with a camera operator or automatically. A camera operator has the ability to direct multiple cameras and follow the speakers in close focus. A simple way to produce a video recording automatically without a camera operator is by fixing a camera with wide focus at the front of the room. We have begun to experiment with automatic person tracking and panoramic cameras (see [10]).

Information about captured meetings can also be accessed from the MBase system. This Web-based system provides access to video collections that are organized into directories grouped by topic (e.g., staff meetings, project reviews, etc.). A number of video analysis techniques are used to give users summaries of and access points into the videos. Figure 9 shows an entry for a video. A timeline visualizes different video features such as camera shots. Moving the mouse over the timeline shows the corresponding keyframes marked by the blue triangles along the timeline. Three icons above the timeline provide links to other applications related to the video or meeting: the first is for Manga (described in section 5 below), the second for LiteSlideViewer to see the captured slide images, and the third for notes taken on pen-based systems with our NoteLook application [6].
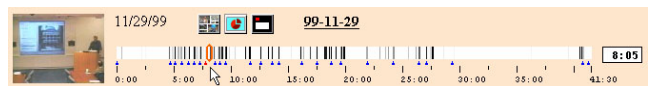

Figure 9. MBase entry for a video. A timeline shows different video features. Icons above the timeline are links to other applications related to the video or meeting.

4.3 Slide Image Capture

Slides are captured by a screen-capture component on the PC workstation whose monitor output is the main display in the meeting room. Images from the display are captured at equal intervals. Captured images are compared to the previous image and saved if a change occurred. There are tradeoffs between the frequency of the capture, the size and format of the saved images, and the load the capturing imposes on the workstation. Capturing, scaling down, and compressing images are all relatively expensive operations that should not interfere with the normal operation of the PC. GIF compression is better than JPEG for the relatively uniform format of slides. Scaling images down before the GIF compression does save time. We determined a color table based on a corpus of color schemes people used in their presentations. We found that capturing images once every 2 seconds, scaling them down to 640x480, and saving them as GIF images without dithering provides sufficient information without interfering too much with the normal operation of the PC. The time that a particular slide is displayed by the presenter is recorded along with the slide image (we encoded the time as part of the file name).

We have used different architectures to capture and use the images. Initially, we created a custom HTTP server that would capture a single screen image and deliver it as a GIF image. That server was contacted periodically by the meeting minutes server using the HTTP If-modified-since field to deal with unchanged images. In that architecture, the meeting minutes server took care of the archival of the screen images. While that architecture provided a nice source for screen images for a number of clients, it did not produce the best performance. We decided to let the screen capture service periodically save the screen images to a directory served by a standard Web server.

We also experimented with instrumenting the Microsoft PowerPoint application to capture all slide change events. While that is not difficult to accomplish, we found that the ease of getting screen images together with the support for presentation applications other than PowerPoint (e.g., Web-based presentations) made our current approach more convenient and flexible.

4.4 Contextual Metadata for Link Generation

Systems that detect and make use of changing environmental conditions are *context-aware* (e.g. see [17]). Context can be used to specify which media are being recorded and which media are being used in a meeting at a particular time. This type of information can be determined by recording when each piece of equipment is switched on or off. We capture and store this information as contextual metadata for the multimedia recording. When links from the text meeting minutes are generated, the metadata determines whether it makes sense to provide the links or not. Examples are illustrated in Figures 4b, 5, and 6, where a [slide] link has been generated only at those times when the speaker showed slides.

Contextual metadata becomes even more important when the device is mobile. Currently, a laptop is used in the one room equipped for meeting capture. For multi-room use, the location of the laptop needs to be known to match the text notes to the captured media streams in that room. One method to do location

sensing is to employ an infrared transceiver system, as in [17]. Another way to locate the laptop is with commercially available GPS, which is somewhat limited for our purpose but can locate the device to the nearest building in a research park or campus. Both of these methods require attaching another device to the laptop.

However, additional hardware attachments are not necessary for a laptop already connected to a wireless LAN system, as recent work at UCLA furnishes a location service that deduces the room location of the device by analyzing the wireless signal (see [3]). This location service has been installed at our lab and we plan to apply it to generate contextual metadata for room location.

4.5 Revision Via Email

The revision process is handled via email (see Figure 10). In the main scenario, after a staff meeting the note taker revises the meeting minutes by editing the contents of the HTML email sent by the LiteMinutes server. The note taker then sends the result either to all lab members (which includes the email address of the LiteMinutes server), or to the appropriate people while cc'ing the LiteMinutes server. In other scenarios, the revision may be performed on a subsequent message from an email thread (not unlike a discussion on a distribution list).
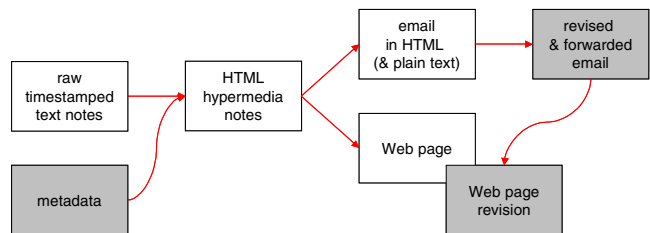

Figure 10. Diagram showing the flow of the text notes to the generated hypermedia meeting minutes, along with email revisions.

The LiteMinutes system handles revisions as follows. LiteMinutes has a mailbox, from which meeting minutes are extracted by searching for sections of email bracketed between special HTML comment tags. The comment tags have the following form, in which the field *Date_Time_IP_Location* is unique for each session taken at a specific time and on a specific laptop at a specific room location:

<!-- LiteMinutes|NotesHtml|Begin|*Date_Time_IP_Location*| -->

<!-- LiteMinutes|NotesHtml|End|*Date_Time_IP_Location*| -->

Extracting the ASCII text between these comment tags filters out emails that are not meeting minutes, and strips off the email headers. The Revisions link on the HTML meeting minutes calls a CGI script that gets a list of revisions. The Weekly Meeting Notes Web page is generated dynamically by a CGI script to link to the latest revisions.

5. VISUALIZATION OF VIDEO AND TEXT CAPTIONS

In addition to a text-centric access to a meeting, we also provide a visual summary that allows users to start video playback at points that look visually interesting. Our video summarization and

visualization system called *Manga* [20] summarizes a video into a comic book style page with different size images, plus text captions in balloons activated when the cursor is over an image (see Figure 11). Image and audio analysis is used to automatically detect events and rate their importance. Keyframes are extracted for the events, with the more important ones shown in a larger size. Manga picks a small subset of the most important events for laying out on a single Web page, with links to video playback from the images.
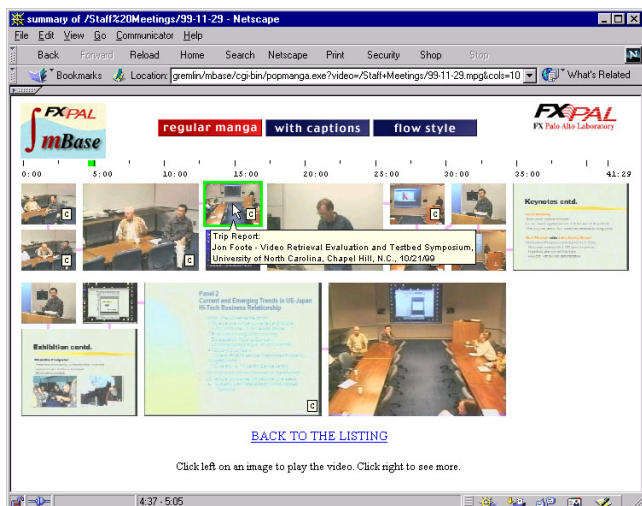

Figure 11. Manga video summarization and visualization system with text captions. Text captions are shown in a tooltip balloon when the cursor is over an image.

The timestamped text notes from LiteMinutes are used to create the Manga captions. Each note item parsed by LiteMinutes furnishes a balloon caption displayed as tooltips in Manga. Before LiteMinutes was developed, Manga captions were created manually while watching the video and aligning the text-only meeting minutes (see Figure 4a) to the right times in the video. Obtaining the text captions automatically from the LiteMinutes data eliminated this painful process.

6. RELATED WORK

There exist more heavyweight and full-featured multimedia note taking systems for meetings. The FXPAL NoteLook system [6] allows users to incorporate images from the video sources of the room activity and presentation material into the notes, and users can take freeform notes with digital ink. It runs on pen-based notebook computers and does not support text notes. The images and ink strokes are indexed to the video recording for retrieval. It requires training to use and is not designed for novices. Other multimedia pen-based systems also require a certain amount of training; examples are Audio Notebook [19], Classroom 2000 [1], FXPAL Dynomite [23], Filochat [22], and Marquee [21].

WEmacs [15] is a text note-taking application based on the GNU Emacs editor. Its user interface is more complicated than the LiteMinutes text box, and it assigns functions to special characters (e.g., a Tab is used to separate note items). Starting and ending a session in WEmacs is also more involved; in contrast, LiteMinutes applet is accessed from the Web and runs inside a Web page. WEmacs serves a purpose different from taking

meeting minutes: its notes are beamed onto a shared display running the Tivoli application on a LiveBoard (an electronic whiteboard) and these notes along with the whiteboard contents plus an audio recording are used to create reports after the meeting [16].

The Where Were We system (W3) [14], which is related to WEmacs, supports making annotations and video recording during a live event. Each note is created in a separate user interface widget, which makes it difficult to use in a live meeting. W3 supports the playing back of the video recording during the meeting itself. W3 does not support slides.

There are a number of related video annotation systems (e.g., see [12] for an overview). A more recent Web-based system is Microsoft MRAS [2]. It is designed for asynchronous video annotation, and supports text and voice annotations. Each text annotation is created in a separate user interface widget. It is based on ActiveX technology, which only works in Windows/Internet Explorer so it is not as portable as Java applets. Users can email text or audio annotation with a single URL, so in effect, each note item is an annotation. This is usable for the task of video annotation, which allows the user to pause the video when making the notes. In contrast, LiteMinutes has a single text box for arbitrarily making note items to support rapid interaction during a live meeting.

There are commercial applications that link timestamped notes to media, such as Souvenir [18]. This product allows users to bookmark audio and video on the Web with their text or handwritten notes taken on a computer, PDA, or CrossPad. Each note item can be played back or emailed to others. While access is Web-based, note creation is not Web-based. It does not provide an easy way to do revisions.

7. CONCLUSION

With LiteMinutes, we have demonstrated how multimedia meeting minutes can be effectively supported on the Internet. The observations of our staff meeting minutes process indicate that a simple text applet on a wireless laptop provides a good way to take notes in a meeting, and that both email and Web access are necessary to satisfy the different preferences for push and pull distribution of meeting minutes. Our architecture of loosely coupled media streams reduces the multimedia capture and access into more manageable modules. We also learned how contextual metadata could be employed to produce cleaner multimedia documents. A phenomenon of the Internet is that boundaries between content and between applications are not always clear-cut, and these interrelations can be exploited as exemplified by our integration of LiteMinutes text notes as captions in our Manga video summarization and visualization system for browsing videos of meetings.

LiteMinutes has been successfully deployed at our lab and has been in use for over 10 months. Because our design carefully considered the pre-existing process, we experienced a smooth transition from the earlier text-only meeting minutes transcribed from notes taken on paper to the LiteMinutes multimedia meeting minutes. In the future, we plan to explore more complex interactions with multiple note takers and more complex spatial arrangements with distributed locations such as teleconferences.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Abowd, G. D., Atkeson, C. G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., and Tani, M. Teaching and learning as multimedia authoring: the classroom 2000 project. *Proceedings of ACM Multimedia '96*, ACM Press, pp. 187-198.

[2] Bargeron, D., Gupta, A., Grudin, J. and Sanocki, E. "Annotations for streaming video on the Web: system design and usage studies," *Proceedings of the Eighth International World Wide Web Conference* (Toronto, Canada, May 1999), available at http://www8.org/. Also in Computer Networks (Netherlands), Elsevier Science, 17 May 1999, Vol. 31, No. 11-16, pp.1139-1153.

[3] Castro P. and Muntz R. Managing context for smart spaces, *IEEE Personal Communications*, August 2000.

[4] Chiu, P., Foote, J., Girgensohn, A., Boreczky, J. Automatically linking multimedia meeting documents by image matching. *Proceedings of Hypertext '00*, ACM Press, pp. 244-245.

[5] Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. Room with a Rear View: Meeting Capture in a Multimedia Conference Room. *IEEE MultiMedia Magazine*, vol. 7, no. 4, Oct-Dec 2000, pp. 48-54.

[6] Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. NoteLook: Taking notes in meetings with digital video and ink. *Proceedings of ACM Multimedia '99*, ACM Press, pp. 149-158.

[7] CrossPad™, A. T. Cross Company. http://www.cross.com.

[8] Cruz, G. and Hill, R. Capturing and playing multimedia events with STREAMS. *Proceedings of ACM Multimedia '94*, ACM Press, pp. 193-200.

[9] Davis, R., Landay, J., Chen, V., Huang, J., Lee, R., Li, F., Lin, J., Morrey, C., Schleimer, B., Price, M., and Schilit, B. NotePals: Lightweight note sharing by the group, for the group. *Proceedings of the CHI '99*, ACM Press, pp. 338-345.

[10] Foote, J. and Kimber, D. FlyCam: Practical Panoramic Video. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2000),* vol. III, pp. 1419-1422.

[11] Girgensohn, A., Boreczky, J., Wilcox, L., and Foote, J. Facilitating video access by visualizing automatic analysis. *Proceedings of Interact '99*, IOS Press, pp. 205-212.

[12] Harrison, B. and Baecker, R. M. Designing Video Annotation and Analysis Systems, *Graphics Interface '92*, pp. 157-166.

[13] Lamming, M. and Newman, W. Activity-based information retrieval: technology in support of personal memory. In F.H. Vogt. (Ed.), *Information Processing '92, Personal Computers and Intelligent Systems*, Vol. 3, pp. Elsevier, pp. 68-81.

[14] Minneman, S. and Harrison, S. Where Were We: Making and using near-synchronous, pre-narrative video. *Proceedings of ACM Multimedia '93*, ACM, New York, pp.207-214.

[15] Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I., and van Melle, B. A confederation of tools for capturing and accessing collaborative activity. *Proceedings of ACM Multimedia '95,* ACM Press, pp. 523-534.

[16] Moran, T. P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., van Melle, W., and Zellweger, P. "I'll get that off the audio": a case study of salvaging multimedia meeting records. *Proceedings of CHI '97*, ACM Press, pp. 202-209.

[17] Schilit, B., Adams, N., and Want, R. Context-aware computing applications. *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, December 1994. IEEE Computer Society.

[18] Souvenir®, i-Recall. http://www.i-recall.com.

[19] Stifelman, L. *The Audio Notebook: Paper and Pen Interaction with Structured Speech*. PhD Thesis. MIT Media Lab, 1997.

[20] Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J. Video Manga: generating semantically meaningful video summaries. *Proceedings ACM Multimedia '99*, ACM Press, pp. 383-392.

[21] Weber, K. and Poon, A. Marquee: a tool for real-time video logging. *Proceedings of CHI '94*, ACM Press, pp. 58-64.

[22] Whittaker, S., Hyland, P., and Wiley, M. Filochat: handwritten notes provide access to recorded conversations. *Proceedings of CHI '94,* ACM Press, pp. 271-276.

[23] Wilcox, L. D., Schilit, B. N., and Sawhney, N. Dynomite: A Dynamically Organized Ink and Audio Notebook. *Proceedings of CHI '97*, ACM Press, pp. 186-193.