Capturing, Structuring, and Representing Ubiquitous Audio

DEBBY HINDUS, CHRIS SCHMANDT and CHRIS HORNER
MIT Media Lab

Although talking is an integral part of collaboration, there has been little computer support for acquiring and accessing the contents of conversations. Our approach has focused on *ubiquitous audio*, or the unobtrusive capture of speech interactions in everyday work environments. Speech recognition technology cannot yet transcribe fluent conversational speech, so the words themselves are not available for organizing the captured interactions. Instead, the structure of an interaction is derived from acoustical information inherent in the stored speech and augmented by user interaction during or after capture. This article describes applications for capturing and structuring audio from office discussions and telephone calls, and mechanisms for later retrieval of these stored interactions. An important aspect of retrieval is choosing an appropriate visual representation, and this article describes the evolution of a family of representations across a range of applications. Finally, this work is placed within the broader context of desktop audio, mobile audio applications, and social implications.

Categories and Subject Descriptors: C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing, H.4.3 [**Information Systems Applications**]: Communications Applications; H.5.1. [**Information Interfaces and Presentation**]: Multimedia Information Systems—*audio input / output*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*interaction styles*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*asynchronous interaction*; *synchronous interaction*

General Terms: Design, Human Factors

Additional Key Words and Phrases: Audio interactions, collaborative work, multimedia workstation software, semi-structured data, software telephony, stored speech, ubiquitous computing

1. INTRODUCTION

People spend much of their workday talking. In Reder and Schwab's [1990] study of professionals, phone calls comprised about 20% of the workday, and face-to-face meetings accounted for an additional 25–50%. Yet, this time

The majority of this work has been supported by Sun Microsystems, Inc
Authors' addresses: D. Hindus, Interval Research Corporation, 1801 Page Mill Road, Building C, Palo Alto, CA 94304; email: hindus@interval.com; C. Schmandt and C. Horner, MIT Media Lab, 20 Ames Street, Cambridge, MA 02139; email. geek@media.mit edu; horner@media.mit.edu.

spent talking has been to a great extent out of reach of computer technology. This loss of speech information is all the more striking, given the dominance of the audio medium in influencing communication outcomes regardless of the presence of visual and other media [Oschman and Chapanis 1974]. Furthermore, speech communication fulfills different communicative purposes than text communication and "is especially valuable for the more complex, controversial, and social aspects of a collaborative task" [Chalfonte et al. 1991, p. 21].

Nevertheless, speech is an underutilized resource for CSCW. Recorded speech has been used in a limited way, in applications that require little organization of the audio data. A number of CSCW systems have focused on synchronous video and audio communication for conferencing, summarized in Egido [1990], and informal communication, including RAVE from EuroPARC [Gaver et al. 1992], Cruiser from Bellcore [Fish et al. 1993], CAVECAT from the University of Toronto [Mantei et al. 1991], PARC's media spaces [Bly et al. 1993], Mermaid from NEC [Watabe et al. 1991], Team Workstation from NTT [Ishii 1990], and COCO from Sun [Isaacs and Tang 1993]. However, the potential to capture these collaborative interactions and use them as a source of data has been rarely exploited to date. Capturing conversations enables repeated hearing of interesting utterances, sharing of conversations with colleagues not present for the original discussion, and collating of conversations with other kinds of communications, such as electronic mail and shared documents, that are already stored on computers.

The Activity Information Retrieval (AIR) project at Rank Xerox EuroPARC illustrates how capture can provide people with access to information about their own previously inaccessible day-to-day activities. Lamming and Newman [1992] make use of EuroPARC's RAVE system that continually videotapes lab members, and they have made the stored video retrievable by using situational information such as where the person was at a particular time (obtained from the "active badges" (developed by Olivetti Research, Limited) worn by laboratory members [Want et al. 1992]) and by using timestamped notations made on pen-based devices during meetings. AIR is, we believe, an example of the progression from support of synchronous interactions to storage and retrieval of the *contents* of interactions.

This article describes various means of capturing speech interactions in everyday work environments; we call this *ubiquitous audio.* Common workday activities other than formal meetings provide a starting point for exploring ubiquitous audio, in terms of both user interfaces and audio processing. Ubiquitous audio can come from a number of sources and through a variety of physical input devices. *Ubiquitous computing* refers to the eventual replacement of explicit computer interactions by specialized smart devices that are unobtrusively present in day-to-day pursuits [Weiser 1991]. The ubiquitous computing approach will eventually lead to sizable but not unmanageable quantities of stored information. For example, assuming four hours of conversation per workday (of which 30% is silence) and 10:1 compression of telephone-quality speech, a year of office speech for one person would require approximately 2 gigabytes of storage.

In many ways, storing and retrieving communication for later review is much more demanding than merely establishing the synchronous channel. Both audio and video are time-dependent media with few sharp boundaries to exploit for indexing or classification. If it were practicable, speech recognition and natural language processing techniques could convert speech to text. Such processing will not be as reliable and as fast as human speech for perhaps decades [Zue 1991], and in any case would cause the loss of nuances carried by the audio signal but not in the words themselves. In the meantime, extending audio technology to spontaneous conversation will require automatic derivation of structure without understanding the spoken words.

Malone et al. [1987] introduced the term "semi-structured messages" in their work on electronic mail. Such messages contain a known set of fields, but some of the fields contain unstructured text or other information. Information Lens users can fill in these fields when writing messages and can write rules to route and sort received messages, based on these additional attributes [Mackay et al. 1989]. We use the term *semi-structured audio* with respect to audio recordings to indicate that some information (e.g., date, time, who is involved in the conversation, and when someone was speaking) about the recordings is known, but the actual words in the recordings are not known. The semi-structured approach defines a framework for making these quantities of audio usable by incorporating acoustical cues, situational data, and user-supplied structure. Acoustical structure includes speech and silence detection and the association of portions of the audio signal with the correct talker. Semi-structure aids in providing flexible access to the data without relying on the explicit creation of structure—users *can* create structure as they see fit, but they are not *required* to create structure for the audio data to be manageable and accessible.

In the following sections, we describe applications for capturing spoken collaboration in work situations and how these applications derive structure during or after capture. We describe user interfaces for later retrieval of these stored speech interactions. Our emphasis is on choosing visual representations and mechanisms for interacting with speech segments that range from seconds-long snippets to hour-long recordings. Finally, we discuss the technological and social contexts of digital audio recording. These contexts include mobile computing devices and the use of speech as a data type across a variety of applications.

2. RELATED WORK

Audio in most CSCW applications is only a medium for synchronous communication and not yet a source of data. Short pieces of recorded speech—speech snippets—are the main use of speech as data in current CSCW applications. These snippets are used in message systems, such as voice mail, and in multiuser editing systems. Speech can also be used to annotate text, a facility demonstrated in Quilt [Fish et al. 1988] and now common in commercial software. The stored speech is not itself structured in these applications; the recorded speech is treated as a single unbroken entity, and the application maintains an external reference to the sound, such as a message number or position within the text. This simple approach is suitable only for snippets and is not very informative for our work.

The Etherphone system at Xerox PARC addressed many aspects of providing a functional interface for stored speech, although annotations of documents were the primary application of stored speech in Etherphone (see Zellweger et al. [1988] for an overview of the Etherphone system). This sophisticated and innovative work included a moving indicator during playback, a sound-and-silence display, segmentation at phrase boundaries, editing, cut and paste of pieces of audio, markers, text annotations, an elegant storage system, and encryption [Ades and Swinehart 1986]. We have replicated many of these features in our work and extended it to explicitly support dynamic displays of conversation and spontaneous capture. We also support lengthy recordings and speech as a data type that can be cut and pasted across a range of applications.

PhoneSlave and HyperVoice are examples of using a semi-structured approach to enrich telephone interactions with respect to messages. PhoneSlave [Schmandt and Arons 1985] used conversational techniques to take a telephone message, asking callers a series of questions and recording the answers. These speech segments could be highly correlated with structured information about the call. For example, the response to, "At what number can you be reached?" contained the phone number. Structured data capture has been applied by Resnick [1992] in HyperVoice, an application generator for telephone-based bulletin boards. HyperVoice applications provide a speech- and touchtone-driven interface that uses the form-entry metaphor. While recording their messages, contributors to the bulletin board are asked to fill in some specific fields using appropriate mechanisms. For instance, the headline field is filled in with a brief recording, whereas expiration dates are given by touchtones so that validity checks can be performed. HyperVoice also supports Resnick's Skip and Skan retrieval mechanism for easily navigating among fields and messages [Resnick and Virzi 1992].

PhoneSlave and HyperVoice demonstrate the value of even simple structuring, and we have taken a similarly simple approach to structure in our applications that capture conversations. A contrasting approach can be seen in hypermedia documents. These embody considerable structure, which is explicitly supplied during the authoring process. Muller and Daniel [1990] implemented HyperPhone, a software environment for accessing voice documents in a conversational fashion. HyperPhone's voice documents are text items that have been structured with links to facilitate access when spoken by a synthesizer. One reported conclusion is that the items must be short and very highly connected for the user interactions to be successful. Arons [1991] describes HyperSpeech, a speech-only hypermedia system utilizing speech recognition for navigation. HyperSpeech nodes contain recorded speech from a series of interviews, and HyperSpeech users can move between topics or between speakers. Hundreds of manually constructed links exist in this system. These examples illustrate the amount of structure needed to make quantities of audio useful. Our work on structuring emphasizes automatic derivation rather than explicit authoring. As we have extended our visual displays and interactions to lengthy recordings, we have had to add features so that applications can support multiple levels of structure.

None of the above applications addresses our primary interest area, spontaneous collaboration. The SoundBrowser project at Apple is the most closely related recent work. A portable prototype for capturing spontaneous user-structured audio has been developed by Degen et al [1992]. They modified a handheld tape recorder so that users could mark interesting portions of recordings of meetings or demarcate items in personal memos. A key point is that these annotations could be made in real time, as the sound was being recorded. The SoundBrowser itself is a Macintosh application for reviewing the stored audio, and it supports innovative visual representations and user interactions, including zooming and scanning operations during playback of the recordings. Although our visual display of recorded audio is quite different from the SoundBrowser's, we have incorporated into our display bookmark-style annotations, zooming, and scanning. We, too, recognized the importance of retrospective marking and invented an interactive dynamic display that continually shows the conversation's recent past.

3. THE "HOLY GRAIL": AUTOMATIC TRANSCRIPTION OF FORMAL MEETINGS

Conspicuously absent from our discussion so far is the notion of capturing the spoken contents of formal group meetings without human transcription. Given the importance of meetings, this is an obvious CSCW application, as indicated by the body of work on electronic meeting systems [Dennis et al. 1988; Mantei 1988]. Due to technological issues, however, it is very difficult to automatically structure recordings of meetings.

One issue is the association of each utterance with a participant. The optimal solution is to record each person's speech on a separate audio channel, but it is quite difficult to get each attendee's speech to be transmitted by only one microphone. In fact, high-quality recordings of meetings are problematic in general, due to background noise, room acoustics, and poor microphone placement with respect to some meeting participants. Using one or more wide-area microphones (such as boundary zone microphones often used for teleconferences) allows more flexibility in seating but compromises audio quality. Highly directional microphones can eliminate some background noise and ambient room noise, but they require careful placement and restrict the mobility of meeting participants. The recording may be intelligible. However, the added noise and variable speech amplitude interfere with further digital signal processing, particularly speech recognition, which is quite sensitive to microphone type and placement.

Transcription of the spoken words is the other issue. Speech recognition of fluent, unconstrained natural language is nowhere near ready yet, even with ideal acoustic conditions. Keyword spotting, a less ambitious approach that could produce partial transcripts, is very difficult when applied to spontaneous speech, especially speech from multiple talkers [Soclof and Zue 1990].

However, word-spotting techniques have been incorporated into an audio indexing and editing system [Wilcox and Bush 1991], and keyword spotting need not be perfect to be useful; the Intelligent Ear [Schmandt 1981] used a graphical color display to indicate the confidence of the word recognition through luminance levels.

4. CAPTURING AND RETRIEVING OFFICE DISCUSSIONS

We have explored ubiquitous and semi-structured audio in a variety of tools that support informal meetings, personal notes, and telephone conversations. This section describes xcapture, one such tool, and issues in structuring the audio it captures. This application is a digital tape loop that provides short-term auditory memory in the office, with no inherent structure to the recording.

4.1 Capturing Office Discussions

When multiple authors are working on a collaborative writing project, one may suggest a new wording to a paragraph, which the other strives to write down but cannot remember. By the time the second author says, "That was perfect, say it again," the words have already been forgotten. Xcapture is meant to supply exactly this sort of short-term memory; because the user remembers the flow of the recent conversation, early versions of xcapture made no direct use of any additional structure in the recorded speech.

Xcapture provides short-term audio memory in the office by making use of workstation recording resources in a background task. Many workstations are now equipped with a speaker and microphone, but in practice the microphone is rarely used. Whenever the microphone is not in use by another application, xcapture records ambient sound into a circular buffer. Five to 15 minutes are typical buffer lengths for the scenario just described; longer discussions could be recorded, but retrieval difficulties make that impractical, as discussed in the next section.

4.2 Retrieving Office Discussions

While recording, xcapture receives digital audio data from an audio server and stores the data in its circular buffer. When the buffer fills, the least recent audio data is discarded and replaced with fresh data. When another application needs to record, the audio server temporarily halts the stream of data to xcapture. During recording, xcapture displays a small animated icon of a moving bar as a reminder that recording is in progress.

Xcapture records until the user clicks on its animated icon, which causes a new window to appear. This window displays the entire audio buffer using a SoundViewer widget, as illustrated in Figure 1. The SoundViewer, used extensively in our work, provides a direct manipulation user interface to recorded audio. Time is displayed horizontally as tick marks. When the user clicks on the SoundViewer, the sound plays and a cursor bar flows into the window from left to right. The mouse can be used to move this cursor and cause the replay to jump to the new location; that is, the mouse provides a time-based offset into the sound, allowing random access.

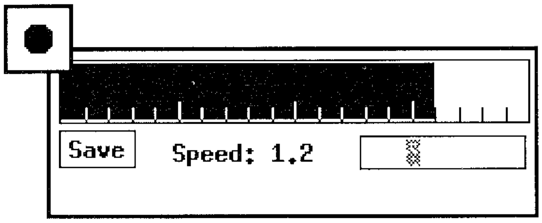


Fig. 1. An early version of Xcapture after an office discussion

During retrieval, the xcapture user wants to find the interesting speech segment within a five-minute long (or longer) recording. The original SoundViewer indicated time, but it gave no clue to the contents of the sound it represented. Retrieval is onerous for a lengthy recording, even with the random-access mechanism described above. Therefore, xcapture and the SoundViewer support scanning through the recorded audio by replaying the speech back in less time than was required to record it.

A slider under the SoundViewer allows the xcapture user to increase playback to up to three times normal speed. Comprehension of the recorded speech is significantly reduced when it is replayed faster than twice normal speed, but scanning through familiar material can be done at even higher playback speeds. Time-compressing speech so that it remains understandable is not straightforward. Simply increasing the rate at which samples are played is inadequate; this raises the pitch, an effect heard in some cartoon characters' voices. Discarding chunks of sound during playback is a better approach; quality is improved by finding and discarding complete pitch periods and by smoothing the boundaries. (For a survey of the techniques and perception of time compression, see Arons [1992a].)

4.3 Discussion of Xcapture

Xcapture works well as a collaborative memory aid during a discussion to immediately replay a consequential utterance or dialog. It is less successful when used to review a conversation; even with random audio access and time-scaling techniques, searching through just ten minutes of audio is tedious. A minor point is that our microphones are battery powered, and we have at times forgotten to turn them on, defeating xcapture's continuous background recording.

Our experience with xcapture led to several research directions to improve the utility of spontaneous recordings. One direction considers how the structure implicit in a conversation can be exploited as part of the capture and retrieval process. A second direction explores improvements to visual representations of stored speech and to audio-related interaction mechanisms. These directions are described in the following two sections, respectively.

5. DYNAMIC CAPTURE AND DISPLAY OF TELEPHONE CONVERSATIONS

This part of our work addresses the derivation of inherent conversational structure, interaction at the time of capture, and the visual presentation of audio during conversations. The inherent structure of a conversation is defined by the naturally occurring pauses at the end of phrases and sentences and by the alternation of utterances between speakers (this alternation is called *turntaking*). The audio data can be automatically segmented into understandable pieces by the Listener, a telephone listening tool that allows users to identify and save the relevant portions of telephone calls as the conversation progresses.

Telephone conversations are a practical choice for demonstrating semi-structured audio; very little equipment is required beyond audio-capable workstations, and talker detection is possible because the two audio channels can be separated. Telephone calls are also typically brief; calls lasted an average of 3–6 minutes in a study of professional work activities [Reder and Schwab 1990].

Studies of audio interactions and telephone calls informed the design of the Listener's segmentation strategy. Beattie and Barnard [1979] focused on turntaking during inquiries to British directory assistance operators. They found that turntaking pauses averaged 0.5 seconds long, although 34% of turns were accomplished within 0.2 seconds. A number of studies quantify conversational parameters, such as turntaking, pausing, and interruptions, as summarized by Rutter [1987]. The above studies suggest that pausing in and of itself is insufficient for structuring a conversation, for three reasons: Turntaking pauses will not be distinguishable from other pauses by their length, not all pauses will be attributable to turntaking, and many turns happen with minimal pausing. The Listener therefore uses both turntaking and the pauses between phrases of one speaker's utterance to derive conversational structure.

5.1 Capturing and Displaying Conversational Structure

The Listener captures structure from telephone calls, as described in the following scenario. You receive a telephone call. The Listener pops up a notification window on your screen. You choose to record the call. While you are talking, a graphical representation of the conversation is constructed on your screen, showing the shifts in who is speaking and the relative length of each turn. You can click on a segment to indicate that it should be saved. At the end of the phone call, you can listen to segments and decide which ones to save, or just save all the marked segments.

Two microphones collect audio signals for the Listener. One is connected to the telephone handset and carries speech from both talkers. The other microphone sits in the user's office near the telephone and carries just that person's speech (assuming that the handset is used rather than a speakerphone). This second, single-talker audio stream enables the Listener to distinguish between the two talkers. The Listener receives audio data from both microphones, performs pause detection on each source, synchronizes the sources, and then locates changes of talker between pauses. This last step is needed because turntaking pauses can be undetectable with just pause detection. The new segment is then added to the call display.

The call display that appears during the conversation must be unobtrusive, so as not to interfere with the conversation. Also, short-term memory constraints imply that only the recent portions of the conversation are salient, and interesting segments can be identified only shortly after the segment takes place.

Figure 2 shows the call display during part of the conversation. As the conversation proceeds, a visual representation of approximately the previous 30 seconds is displayed retrospectively. Each conversational turn is shown, reflecting the phrase-level utterances of the talkers. Each segment, or portion of the audio signal, is displayed within a SoundViewer, the same audio representation that is used throughout our applications. In this picture, each tick mark within each SoundViewer represents one second of audio. New segments appear at the right-hand side, and older segments scroll out of view to the left. Segments from each talker can be visually distinguished by their relative positioning and by different border colors when unmarked.

5.2 Adding User-Supplied Structure

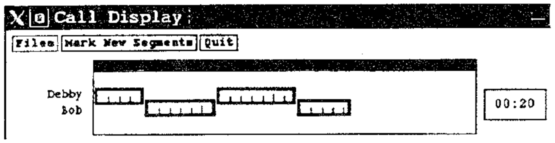
The Listener provides mechanisms for users to mark segments of the conversation that are interesting and may merit later rehearing. The Listener's visual display of conversational structure reinforces the user's memory of significant conversational segments and allows users to identify those segments. A user can mark individual segments at any time by clicking on them. Marked segments are visually highlighted, and marking is reversible. Another marking mechanism requires even fewer user interactions; when the conversation turns to substantial matters, the user can toggle automatic marking so that all new segments are marked.

During the phone call, the user's attention is focused on the conversation and not on interacting with the Listener program. Therefore, the only feasible user actions are clicking the pointer on segments of interest or on the automatic marking toggle. Once the conversation is completed, the nature of the user interaction changes from capture to review. The postcall Browser application displays the entire conversation and provides additional editing functions. A user can replay all or part of the conversation, revise the choice of segments to store, and save the segments for later retrieval. Users can also provide a descriptive tag for each conversation, although tags are not required.

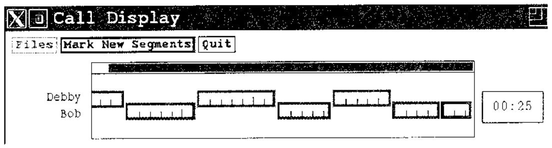
Once these postcall revisions are made, only the marked segments are saved. Marked segments will typically occur in consecutive groups, and when the conversation is retrieved in the future these groups are visually distinct, as shown in Figure 3.

5.3 Retrieving Stored Telephone Conversations

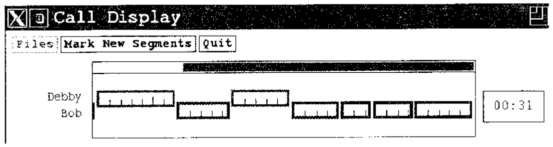
Stored conversations may be retrieved long after the phone call took place. Situational and supplemental structure can provide memory cues to the



D: Hello, this is Debby Hindus speaking.
B: Hi Deb, it's Bob. I'm just getting out of work, I figured I'd call and see how late you're going to stay tonight.
D: Well, I think it'll take me about another hour, hour and a half, to finish up the things I'm doing now.
B: OK, I'm just going to head on home, I'll probably do a little shopping on the way.



D: Well, if you think of it, maybe you could get some of that good ice cream that you got last week.
B: OK. By the way, somebody, uh...
B: mentioned an article you might be able to use



B: in your tutorial. Debby: Oh really? [Debby's very short turn is ignored.]
B: Yeah, it's by Graeme Hirst, in the June '91 Computational Linguistics.

Fig. 2. Sequence of segments during a phone call, with transcriptions.

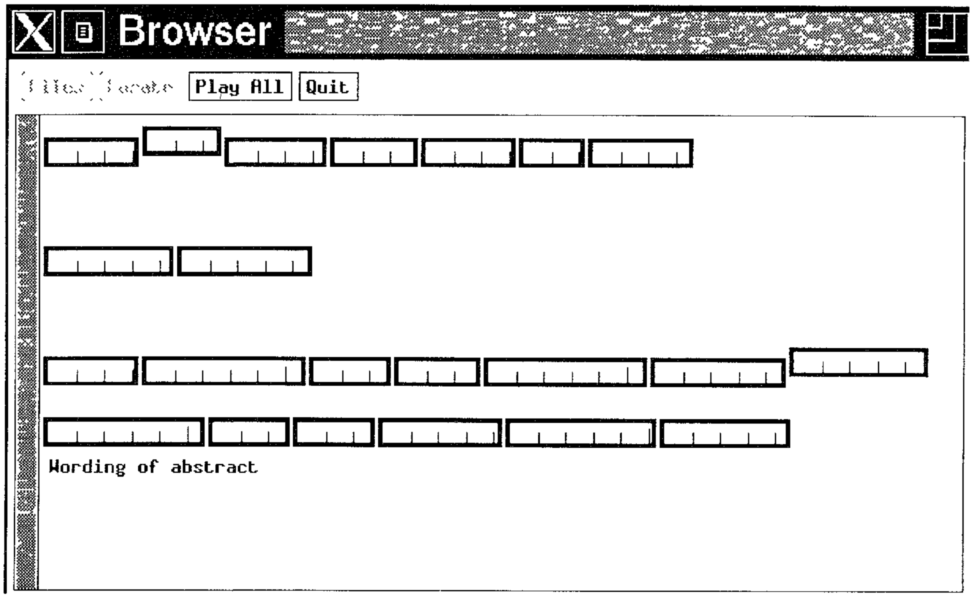


Fig. 3. Browsing a stored conversation with three groups of saved segments.

content of the stored audio; speech cannot be searched like text. The Listener collects and stores situational information; for telephone calls, situational data includes the other party's name and phone number, if known; the time and date of the call; and the phone number of the user. The user's choice of which segments to save is one form of supplemental structure, and textual tags are another. The representation of the marked segments and indices into the corresponding audio data, along with situational, conversational, and supplemental structure, is stored in a file and referred to as a *chat*.

There are two kinds of chat retrieval: one is finding the desired audio segments within a chat, and the other is locating a particular chat from among numerous stored chats. Our work has been narrowly focused on capturing and retrieving segments within a single conversation. Future efforts will need to address mechanisms for navigating among many chats, such as making use of situational data to locate a chat in a fashion akin to locating an electronic mail message.

5.4 Discussion of the Listener

We have used the applications ourselves enough to be confident that the underlying concepts are viable and worthy of additional research. We have, for example, used the Listener while collaborating long-distance on papers and mailed xcapture recordings of impromptu office discussions to other group members. Although the Listener's day-to-day usage was limited to one of the authors for technical reasons, that author experienced consistent success in marking segments of interest or adjoining segments. Furthermore, the minimal interactions did not noticeably interfere with her participation in conversations, and the dynamic display was engaging and comprehended by casual observers. The Browser's static display was considerably less understandable.

Experience with the Listener highlights several aspects of building real-time interactive applications involving conversation. One continuing problem is how to obtain consistent high-quality audio from microphones in offices. The audio quality from telephones is good, but using two microphones to segment based on talker is awkward and imperfect. Another aspect is how to divide the audio signal into segments. Final segment determination must reflect constraints, such as a minimum length of two seconds, that ensure visually distinguishable segments. Additionally, segments should sound complete when played individually or sequentially. As shown in Figure 4, segment boundaries are calculated by the Listener so that they fall in the pauses between utterances.

Another significant aspect is how well the chosen visual presentation works during and after the conversation. The Listener's call display does represent the conversational structure of speech, and it worked well as a dynamic representation during the conversation. It was less successful as a static representation for later browsing. Clearly, there are opportunities for experimentation and innovation with respect to representation and interaction, particularly when informed by the cognitive science perspective. For example, interacting with a computer program while engaged in conversation raises issues of task and memory workload, and use of attentional resources.

Finally, privacy issues received only minimal attention in this prototype implementation. As we discuss toward the end of this article, applications that record conversation need to accommodate privacy concerns before they can be employed outside of a small research group.

6. PRESENTING AND INTERACTING WITH LENGTHY RECORDINGS

We saw when working with xcapture that the original SoundViewer did not accommodate lengthy recordings well, and it proved to be too simple for audio material with application-specific or user-supplied structure. It worked well for speech snippets, however, and the Listener avoided the SoundViewer's limitations by using SoundViewers for each segment of the conversation and arranging them to represent conversational structure. In this section, we describe enhancements we made to the SoundViewer that enable it to directly support segmentation, multiple levels of structure, and presentation of lengthy recordings. These enhancements include the display of segmentation, scaling and zooming of long sounds, and the ability to annotate parts of the sound with text or markers that act as bookmarks. Mechanisms for navigating among segments and for rapid searches were developed as well.

6.1 Displaying Multiple Levels of Structure

The enhanced SoundViewer widget supports the optional display of several levels of structure. The most general structuring for speech is to distinguish

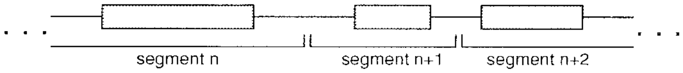


Fig. 4 Silences are divided between segments.

between speech and silence intervals. Figure 5 shows the modified widget incorporated into a voice mail application; segments of speech are displayed as black bars, and silence is white, following Etherphone's example. The SoundViewer allows the user to jump forward and backward between speech segments during playback, by pressing the space bar or "b" key, respectively.

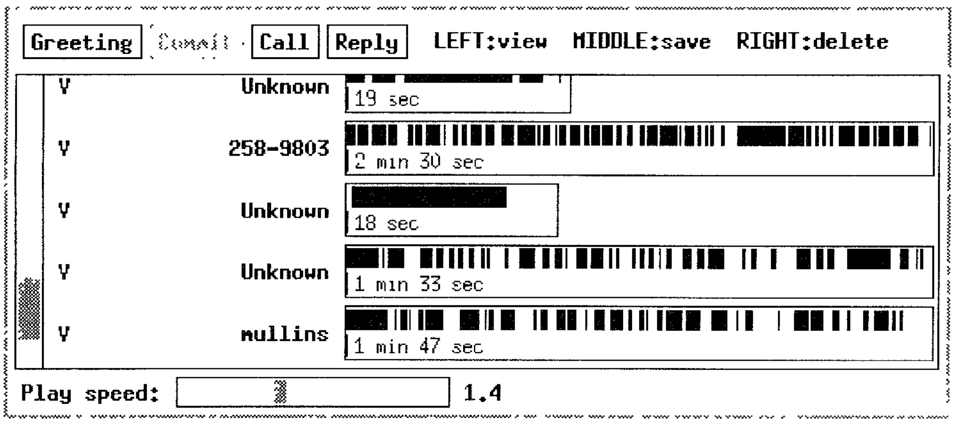
Applications may require segmentation at a level of semantic and structural information higher than speech and silence, such as distinguishing between two speakers or between music and speech. To present this application-specific level of structure, an application containing a SoundViewer may specify its own layer of black and white bars below the speech and silence bars, as shown in Figure 6. The interpretation of this content layer is up to the application, which can also associate a visual icon with the content. For example, musical interludes in a radio show are indicated by musical notes. Keystrokes can be used to skip from one content bar to another.

The third level of structure is user-supplied structure. Users can denote points of interest within a recording by adding visual bookmarks. We followed the example of the SoundBrowser [Degen et al. 1992]; users can place arrow-shaped markers in the SoundViewer by pressing the caret key. Text is another way to convey information within a SoundViewer, and an optional text label can be set by the application. This label can display the caller's name in a telephone message, for example, or the date of a recording.

6.2 Displaying and Interacting with Lengthy Recordings

The SoundViewer emphasizes the temporal aspect of audio together with interactive playback controls and has required improvement to accommodate recordings longer than a few minutes. Like other graphical interfaces to time-varying media, the SoundViewer uses a mapping from time to space (length) to represent sound. Showing the total duration of a recording, along with a continually updated position indication during playback, is important for navigation and user comfort [Myers 1985].

The SoundViewer initially used tick marks of varying size to convey a time scale, and longer sounds were shown with closer spacing between tick marks. But these visual cues were inadequate for indicating total duration or positioning within long recordings. Because tick marks failed to present absolute duration, text labels were introduced into the SoundViewer to display sound duration; for example, "1 min 30 sec" labels a 1.5-minute recording. Tick marks did provide navigational cues, however, and we are evaluating tick marks and speech-and-silence displays for navigation.



V Unknown 19 sec

V 258-9803 2 min 30 sec

V Unknown 18 sec

V Unknown 1 min 33 sec

V mullins 1 min 47 sec

Play speed: 1.4

Fig. 5. Enhanced SoundViewer in Vmail, showing speech and silence intervals.

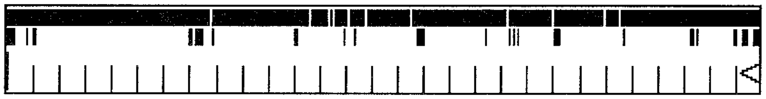


Fig. 6. Content bars displayed beneath segmentation layer.

Accurate position indication is important for using the time-to-space mapping to support direct manipulation during playback. But when recordings become very long (e.g., a thirty-minute conversation), so much audio corresponds to a single pixel that this mapping breaks down; the bar moves so slowly during playback that the display appears static and provides inadequate feedback. Additionally, if the user chooses to move to a different location within the sound, temporal resolution is too coarse.

To overcome the SoundViewer's resolution limitations, an interface layer called MegaSound was added above the SoundViewer. Following the Hierarchical Video Magnifier of Mills et al. [1992], MegaSound can expand and compress pieces of the timeline without losing global positioning information. A MegaSound widget, shown in Figure 7, consists of two SoundViewer widgets, with lines connecting them, that show the zoom region in its global context. Because MegaSound incorporates SoundViewers, SoundViewer behaviors are preserved, such as speed control and random access, by moving the position indicator. The user can interact with either the root level or the zoomed SoundViewer, and the region of magnification moves in synchrony with playback. MegaSound maintains the link between the two layers so that the position indicators are synchronized.

MegaSound also generalizes the bookmark mentioned earlier, by turning it into an annotation where the user can type arbitrary text. It does this by "flagging" bookmarks and linking them to a text entry window. The annotation text can be entered by the user or can be provided by the application. As

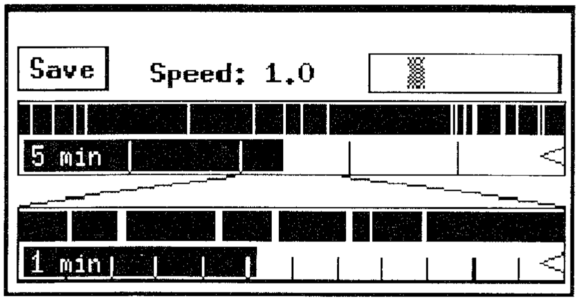


Fig. 7. Revised xcapture, showing a MegaSound widget with both root and zoom levels.

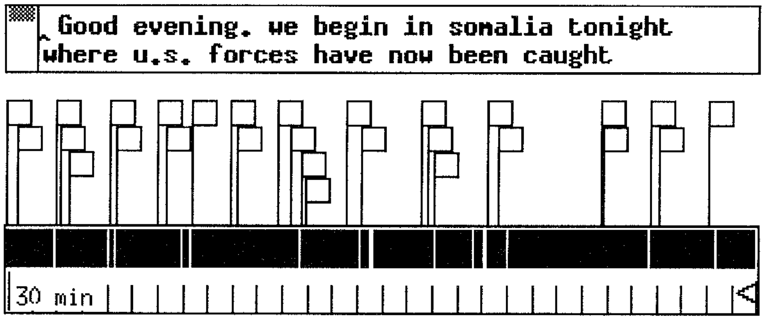
the pointer is moved into a given flag, its associated text is automatically displayed in the text window, and the sound's position indicator automatically jumps to the flag's position, as shown in Figure 8. Clicking on a flag sets the zoom region of the MegaSound widget to the temporal boundaries for that flag and plays the associated audio. Accelerator keys can also be used for navigating among the annotations.

Along with adding the MegaSound layer, the SoundViewer itself was improved with respect to scanning and speed control. Playback speed can be specified with the numeric keys or changed relative to the current speed with the plus and minus keys. Additionally, the SoundViewer includes speed control as part of the scanning interface. Mouse motion events are treated as play commands; as the user clicks and drags the position indicator forward or backward, there is accompanying audio feedback. The faster the dragging motion, the faster the audio plays in order to cover the region swept out.

7. IMPLEMENTATION OVERVIEW

The audio applications described in this article were developed on a common platform and were designed to complement each other. Software was developed in the C language on a Sun Sparcstation 2 running SunOS. Releases 4 and 5 of the X Window System toolkit were used with the Athena (Xaw) widget set. The Sparcstation 2 can digitize a single sound channel at a sampling rate of 8,000 samples per second in an 8-bit μ-law format, which is comparable to a good-quality telephone connection; a 1-minute recording is 480KB long without further compression.

The desktop applications described here are X Window System client applications that rely on independent servers—the X Server and the audio server—to handle their interactions with the user's display and workstation audio, respectively. The workstation's X server displays the user interfaces and passes along user input events. The audio server is a networked server for managing asynchronous audio operations that is built on top of audio library routines. (These device-independent routines provide a level of abstraction above the workstation audio devices and support nonblocking



Good evening. we begin in somalia tonight
where u.s. forces have now been caught

Fig. 8. A nonzoomed MegaSound widget with annotation flags.

audio operations.) The audio server's architecture emulates that of the X Window System in that it consists of a server process that executes on the workstation and a client-side library of routines that are linked into an application program. The client-side routines enable the application to communicate with the server process through callbacks [Arons 1992b].

The Listener also made use of the Phoneserver, another independent server. The Phoneserver monitors the status of the group's ISDN telephone lines and can deliver phone-related events to programs such as the Listener. The Phoneserver currently resides on a Sun Sparcstation with built-in ISDN telephone interface hardware.

To implement the Listener, the ChatViewer widget was created. The ChatViewer is an X Window System widget that displays and keeps track of two kinds of items: the sound segments generated by the Listener and text strings added by users. It makes use of two other components, the SoundViewer widget and the database manager. The database manager provides a simple record-oriented database in which fields are defined as key-value pairs. ChatMan (short for "Chat Manager") was designed as a more general superclass of the ChatViewer widget. ChatMan manages the item database, just like the ChatViewer, but leaves all widget decisions up to its subclass. Among these choices made by the subclass are how to lay out items and what type of widget to use for sound or text items. The OmniViewer is one such subclass, so named because it can handle all possible layouts by acting as a bulletin board. (See Hindus [1992] for lower-level descriptions of ChatViewer functions and implementation details, and Horner [1993] for detailed descriptions of the ChatMan, OmniViewer, and MegaSound widgets.)

8. BEYOND CAPTURE, STRUCTURE, AND PRESENTATION

Although xcapture and the Listener are worthwhile by themselves, such ubiquitous audio applications are more powerful when implemented in a broader computing context. This section considers three contexts in which these applications operate. The first context is a family of applications employing speech at the desktop. These applications feature a consistent visual interface and interoperability. The second context is portability and mobility to allow speech to be captured in a wider range of social situations. The final context is the social implications of ubiquitous recording technology.

8.1 Desktop Audio Applications

The SoundViewer and related audio widgets are used in a number of other Media Lab audio applications. As shown in Figure 9, the various audio applications make use of shared text and audio databases regardless of whether the user's interactions are graphical or telephone based.

A screen-based user interface to voice mail, shown in Figure 5, provides random access within and between messages, speed control of playback, and call-return and message-return capability [Stifelman 1991]; messages can originate from unanswered calls, other voice mail subscribers, or audio attachments to ordinary Internet email. Speech snippets can be used as daily calendar entries in xcal, a speech and text personal schedule [Schmandt 1990]. A speech editor, Sedit, uses the speech-and-silence display to indicate the extent of phrases that the user can cut, paste, and augment with additional recording.

NewsTime provides a visual interface to broadcast radio news and to digital audio recordings delivered over computer networks (Internet Talk Radio), and it utilizes the structure inherent in news programming [Horner 1993]. Figure 10 shows the NewsTime application. (In this picture, the right-hand window displays MegaSound widgets for four feature stories; the top story is being played, so the top MegaSound is expanded to magnify the minute-long region of the recording that the user is hearing. The left-hand window displays the text summary of the talk show.)

A summary paragraph does not necessarily correspond to a single MegaSound, although both the text and the recording are in order.

The use of the SoundViewer in all these applications provides audio cut-and-paste capabilities between applications in addition to a consistent visual user interface. The SoundViewer supports cut and paste through standard X Window System selection mechanisms. A segment of audio can be selected in any SoundViewer and then pasted into any other application that supports recording, thus increasing the utility of simple capture applications. For example, one could take a portion of a conversation and send it as voice mail to a third party who was asked to comment. Or one might agree to perform a task and then copy the portion of the conversation describing the task into one's calendar on the task due date, as shown in Figure 11.

Tools to capture ubiquitous audio contribute a new source of audio as a data type to be shared and manipulated by a wide range of desktop applications.

8.2 Mobility

Xcapture and the Listener provide recording of office conversations. Productive conversations also occur spontaneously in hallways, common areas, and outside of work sites. Capture applications will ultimately employ technolo-

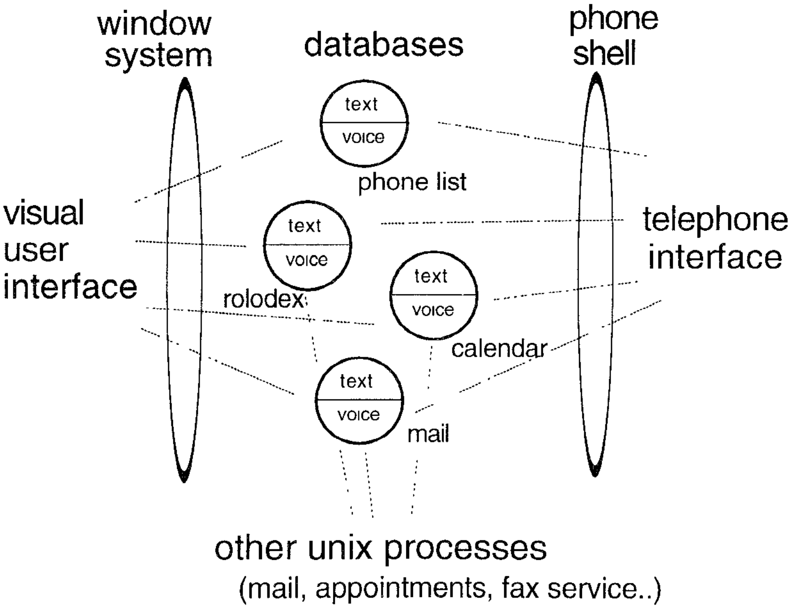
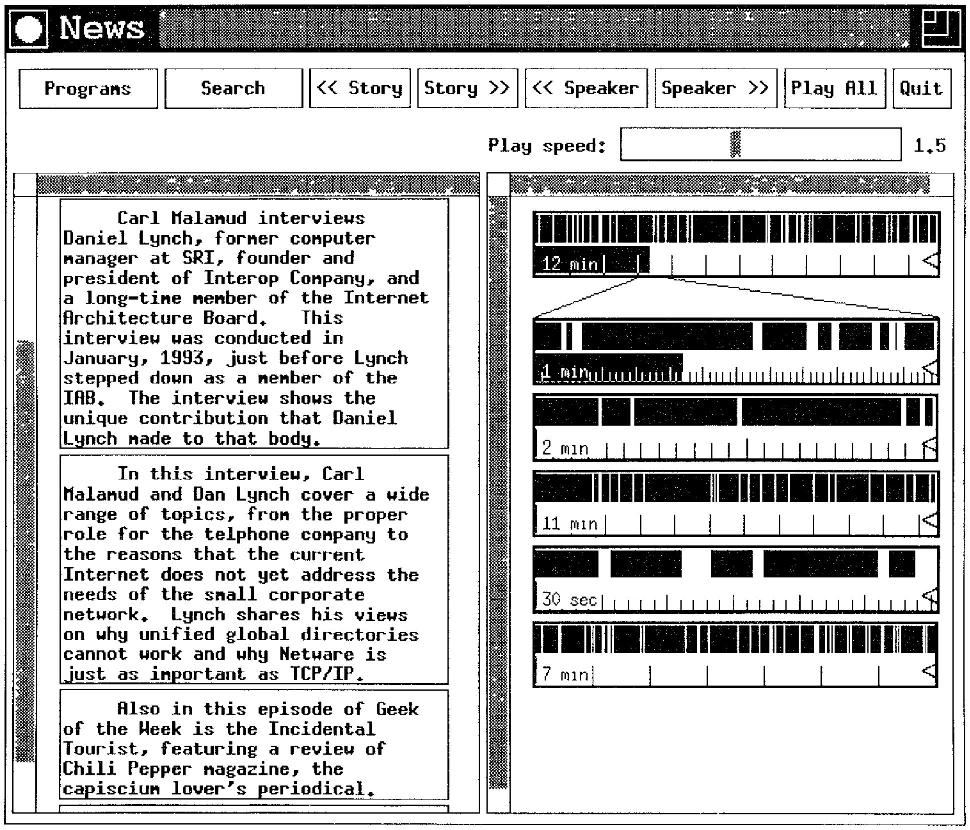


Fig. 9. Multimedia databases are accessed by applications that support various media for presentation.

gies that can operate in such environments as well. The Media Lab's initial portable prototypes and telephone-based user interfaces show how such ubiquity might be achieved.

A prototype hand-held voice computer supports speech recognition for accessing VoiceNotes, Stifelman et al.'s digital audio file system [Stifelman et al. 1993]. VoiceNotes users record memos into categories; each category consists of a list of memos that can be accessed sequentially. By eliminating the keyboard and display, portable devices could fit in a pocket and literally always be at hand. Speech interfaces also allow interaction while the user's hands and eyes are busy—while driving, for example, or in dimly lit environments. VoiceNotes also provides a nonvisual version of xcapture for recording conversations. During recording, speech is segmented based on pauses, and these segments become sequential elements of a list that can be browsed under speech or manual control, or saved to other VoiceNotes audio files [Stifelman 1992].

Although these Media Lab hand-held prototypes are tethered to a laptop computer, telephone-based personal information management applications offer the complete mobility afforded by hand-held cellular telephones. Phoneshell is a family of such applications, using digitized and synthesized speech to provide access to voice mail, email, calendar, and personal name and telephone number databases, as well as news, weather, and traffic. Phoneshell users can record messages into databases shared with the desktop speech applications described above, or type short text entries such as "ok" on



Carl Malamud interviews Daniel Lynch, former computer manager at SRI, founder and president of Interop Company, and a long-time member of the Internet Architecture Board. This interview was conducted in January, 1993, just before Lynch stepped down as a member of the IAB. The interview shows the unique contribution that Daniel Lynch made to that body.

In this interview, Carl Malamud and Dan Lynch cover a wide range of topics, from the proper role for the telphone company to the reasons that the current Internet does not yet address the needs of the small corporate network. Lynch shares his views on why unified global directories cannot work and why Netware is just as important as TCP/IP.

Also in this episode of Geek of the Week is the Incidental Tourist, featuring a review of Chili Pepper magazine, the capiscium lover's periodical.

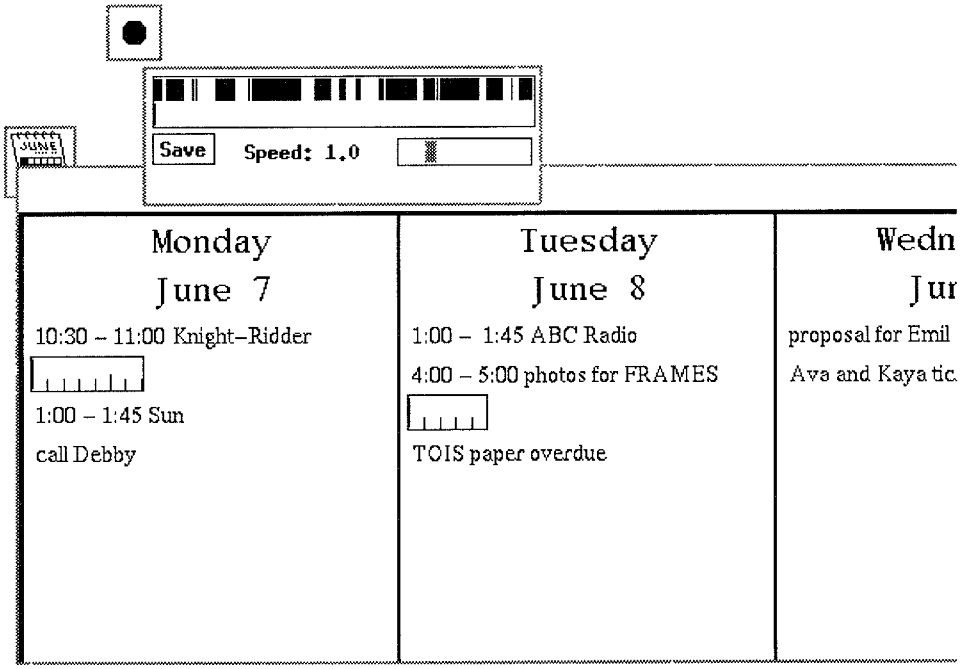
Fig. 10. NewsTime application displaying a portion of an Internet Talk Radio talk show.

the telephone keypad. Phoneshell has been operating for over two years and is regularly used by group members [Schmandt 1993].

Recorded speech that was captured by mobile devices can be retrieved at the desktop. For example, the voice mail system offered under Phoneshell provides a top-level menu function to record personal memos. (This function is routinely used by one of the authors to record ideas during long walks or to note tasks on the commute home.) These memos are automatically transferred to a free-form speech and text display, as shown in Figure 12. Here, speech and text items can be sorted and arranged spatially into different categories.

8.3 Social Context and Privacy

Ubiquitous audio will exist within the social context of individuals and organizations. The applications described in this article can clearly benefit the individual user, as we have experienced within a small, trusted group of researchers/users. Outside of that limited context, the social impacts are not all beneficial. Capturing conversation has social impact because it challenges



Monday
June 7
10:30 – 11:00 Knight-Ridder
1:00 – 1:45 Sun
call Debby

Tuesday
June 8
1:00 – 1:45 ABC Radio
4:00 – 5:00 photos for FRAMES
TOIS paper overdue

Wedn
Jur
proposal for Emil
Ava and Kaya tic

Fig. 11. Recorded speech can be selected from xcapture and pasted into the calendar.

an ingrained assumption that conversation is ephemeral. Sproull and Kiesler [1991], in discussing electronic mail as ephemeral, describe ephemeral communication as marked by a lack of tangible artifacts. They remark that ephemeralness causes people to be less committed and concerned about what they say and how it will be received. Xcapture and the Listener enable conversation to become tangible and storable; therefore, conversation will become less ephemeral. This change in assumptions makes it important to take into account privacy concerns.

Bellotti and Sellen [1993] have proposed a framework for designing ubiquitous systems that incorporate privacy mechanisms. The framework defines control and feedback as the key considerations that must be addressed for four system or user actions—capturing data, constructing representations of the data for storage, accessibility of stored data, and the purposes to which stored data can be put. As Bellotti and Sellen point out, feedback and control over capture are the most important issues for privacy. One issue is feedback during capture, or notification—people do not like to feel spied upon. For speech capture tools, an audible signal is most appropriate, particularly if it directly conveys the connotation of a recording [Gaver et al. 1992].

Another issue relates to control: consent to the capture during the conversation. Some video-based systems have incorporated symmetry; you see me only if I see you. Often a telephone call model is employed, in which video requests for communication are accepted or rejected. Strict symmetry, while

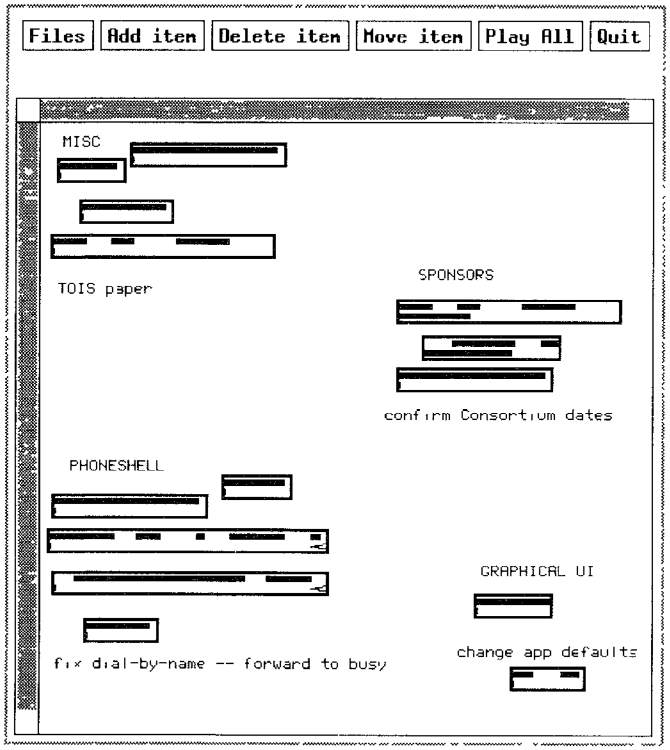


Fig. 12. Spatial layout into categories supports recall in a mixed voice and text project management list

suitable for some kinds of video interaction, would overly limit the utility of a telephone-based capture system such as the Listener—only calls from people at Listener-capable workstations could be captured. Consent on the part of someone within the group could be negotiated using access controls based on who and how they were calling, and interactive controls, such as in the RAVE [Dourish 1993] and Cruiser systems [Fish et al. 1993]. Obtaining consent from remote collaborators outside of the workplace is more problematic. A touchtone interaction confirming consent might work, although conversants may well need to be reminded periodically of this option during lengthy conversations.

Making conversation retrievable raises issues of the purposes for which the stored material might later be used. The general questions of the risks of stored information and guidelines for how stored digital data can be safeguarded have been addressed in Dunlop and Kling [1991] and Rothfeder [1992]. Potential misuses of audio capture systems include later eavesdropping by third parties, editing to misrepresent what was said, and even examination for political attitudes long afterward. One barrier to unintended use is encryption of the stored recordings. Encryption during the conversation would also be a barrier to real-time snooping. Another idea is to build in autodestruct mechanisms, so that routine conversations, or perhaps ones that have not been accessed since recording, are removed periodically. This does not eliminate their existence on backup copies, however.

Organizational and political forces may supersede an individual's choices, but systems should support those choices in the meantime, such as providing encryption and affordances for privacy. The Bellotti and Sellen framework shows how to design those affordances, and Dourish's Godard, a software infrastructure for providing flexible control and feedback mechanisms in computer-mediated communication systems, illustrates how these affordances can be implemented [Dourish 1993].

9. CONCLUSION

This article has introduced the concept of ubiquitous gathering of audio and presented applications and user interfaces to manage ubiquitous audio. We have only begun to explore capturing, structuring, and presenting large amounts of audio, and we expect to continue this research. Follow-on work at Interval Research will include exploring visual presentations of audio and designing ways to make audio more tangible to users. The Media Lab is pursuing auditory presentation techniques for skimming through audio [Arons 1993] and audio interactions for accessing news databases, in addition to continuing the development of the audio applications described above.

Another direction for future work is in new approaches to deriving structure. A promising approach is to delineate segments of interest by examining prosody; Chen and Withgott [1992] exploited prosodic information in creating an emphasis detector that could automatically locate segments of discourse that listeners found to be significant and that could operate on telephone-quality speech. Other areas of interest include extending capture applications to meetings of more than two persons and exploring the technical and social aspects of portable, ubiquitous recording devices.

The work described in this article demonstrates that everyday speech interactions can be captured in various ways, and semi-structuring the audio data aids in storage and retrieval. Real-time structuring and display is feasible by making use of situational constraints and by adopting simple segmentation strategies, and lengthy recordings can be presented visually. These concepts, along with the related issues of privacy, mobility, and audio as a data type, will influence the value and utility of recorded speech as audio becomes more ubiquitously available in workday situations.

ACKNOWLEDGMENTS

The work presented in this article relies on software developed by members of the Media Lab Speech Research Group. Barry Arons designed and built the audio server, a key component of the audio software. Both he and Lisa Stifelman contributed to the research content and the software engineering of many of the audio applications. The SoundViewer widget was initially written by Mark Ackerman, who also provided invaluable training and advice on X Window System applications and widget writing. The mechanics for audio cut and paste were implemented by Sheldon Pacotti; xcapture was written by Lorin Jurow; and Eric Ly and Atty Mullins worked on related audio applications.

We are indebted to Tom Malone for his persistent, thoughtful guidance and support in the development of this article, and to the anonymous reviewers for their significant and careful comments. We also thank Diane Schiano and James Baker for their critiques; Terry Winograd and Amy Bruckman for their contributions to the discussion on privacy issues; and Mark Ackerman and Wendy Mackay for their comments on an earlier version of this article.

REFERENCES

ADES, S., AND SWINEHART, D. C. 1986. Voice annotation and editing in a workstation environment. In *Proceedings of the 1986 Conference.* The American Voice I/O Society, San Jose, Calif., 13–28.

ARONS, B 1993. Interactively skimming recorded speech In the *Symposium on User Interface Software and Technology—UIST'93 Conference Proceedings.* ACM, New York.

ARONS, B. 1992a. Techniques, perception, and applications of time-compressed speech. In *Proceedings of the 1992 Conference.* The American Voice I/O Society, San Jose, Calif., 169–177.

ARONS, B 1992b. Tools for building asynchronous servers to support speech and audio applications. In the *Symposium on User Interface Software and Technology—UIST'92 Conference Proceedings.* ACM, New York, 71–78.

ARONS, B. 1991. Hyperspeech Navigating in speech-only hypermedia. In *Hypertext '91* ACM, New York, 133–146.

BEATTIE, G. W., AND BARNARD, P. J 1979. The temporal structure of natural telephone conversations (directory enquiry calls) *Linguistics 17*, 213–229.

BELLOTTI, V., AND SELLEN, A. 1993. Design for privacy in ubiquitous computing environments. In *Proceedings of European Conference on Computer Supported Cooperative Work.* Available as Rank Xerox EuroPARC Tech Rep EPC-93-103

BLY, S. A., HARRISON, S. R., AND IRWIN, S. Media Spaces: Video, audio, and computing. *Commun. ACM 36*, 1 (Jan.), 28–46.

CHALFONTE, B L, FISH, R S, AND KRAUT, R. E 1991. Expressive richness: A comparison of speech and text as media for revision. In *Human Factors in Computer Systems—CHI'91 Conference Proceedings.* ACM, New York, 21–26.

CHEN, F. R., AND WITHGOTT, M M. 1992 The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* IEEE, New York, I-229-232.

DEGEN, L., MANDER, R., AND SALOMON, G. 1992. Working with audio: Integrating personal tape recorders and desktop computers. In *Human Factors in Computer Systems—CHI'92 Conference Proceedings.* ACM, New York, 413–418.

DENNIS, A. R., GEORGE, J. F., JESSUP, L. M., NUNAMAKER, J. F., JR., AND VOGEL, D. R. 1988. Information technology to support electronic meetings. *MIS Q. 12*, 4, 591–624.

DOURISH, P. 1993. Culture and control in a media space In *Proceedings of the European Conference on Computer Supported Cooperative Work.* Available as Rank Xerox EuroPARC Tech Rep. EPC-93-101

DUNLOP, C., AND KLING, R., EDS. 1991. *Computerization and Controversy: Value Conflicts and Social Choices.* Academic Press, New York.

EGIDO, C. 1990 Teleconferencing as a technology to support cooperative work· Its possibilities and limitations. In *Intellectual Teamwork· Social and Technological Foundations of Cooperative Work.* Lawrence Erlbaum, Hillsdale, N.J., Chapter 13, 351–371.

FISH, R. S., KRAUT, R. E., LELAND, M.D., AND COHEN, M. 1988 Quilt: A collaborative tool for cooperative writing. In *Conference on Office Information Systems—COIS'88 Conference Proceedings.* ACM, New York, 30–37.

FISH, R., KRAUT, R., ROOT, R., AND RICE, R. 1993. Video informal communication. *Commun. ACM 36*, 1 (Jan.), 48–61.

GAVER, W., MORAN, T., MACLEAN, A., LOVSTRAND, L., DOURISH, P., CARTER, K., AND BUXTON, B. 1992. Realizing a video environment: EuroPARC's RAVE system. In *Human Factors in Computer Systems—CHI'92 Conference Proceedings.* ACM, New York, 27–35.

HINDUS, D. 1992. Semi-structured capture and display of telephone conversations. Master's thesis, Massachusetts Institute of Technology, Cambridge, Mass.

HORNER, C. 1993. NewsTime: A graphical user interface to audio news. Master's thesis, Massachusetts Institute of Technology, Cambridge, Mass.

ISHII, H. 1990. TeamWorkStation: Towards a seamless shared workspace. In *Computer Supported Cooperative Work—CSCW'90 Conference Proceedings.* ACM, New York, 13–26.

ISAACS, E. A., AND TANG, J. C. 1993. What video can and can't do for collaboration. In the *1st International Conference on Multimedia.* ACM, New York, 199–206.

LAMMING, M., AND NEWMAN, W. 1992. Activity-based information retrieval: Technology in support of human memory. Tech. Rep. 92-002, Rank Xerox EuroPARC.

MACKAY, W. E., MALONE, T. W., CROWSTON, K., RAO, R., ROSENBLITT, D., AND CARD, S. K. 1989. *How do experienced Information Lens users use rules*? In *Human Factors in Computer Systems—CHI'89 Conference Proceedings.* ACM, New York, 211–216.

MALONE, T. W., GRANT, K. R., LAI, K.-Y., RAO, R., AND ROSENBLITT, D. 1987. Semi-structured messages are surprisingly useful for computer-supported coordination. *ACM Trans. Office Inf. Syst. 5*, 2, 115–131.

MANTEI, M. 1988. Capturing the Capture Lab concepts: A case study in the design of computer supported meeting environments. In *Computer Supported Cooperative Work—CSCW'88 Conference Proceedings.* ACM, New York, 257–270.

MANTEI, M., BAECKER, R., SELLEN, A., BUXTON, W., AND MILLIGAN, T. 1991. Experiences in the use of a media space. In *Human Factors in Computer Systems—CHI'91 Conference Proceedings.* ACM, New York, 203–208.

MILLS, M., COHEN, J., AND WONG, Y. Y. 1992. A magnifier tool for video data. In *Human Factors in Computer Systems—CHI'92 Conference Proceedings.* ACM, New York.

MULLER, M. J., AND DANIEL, J. E. 1990. Toward a definition of voice documents. In *Conference on Office Information Systems—COIS'90 Conference Proceedings.* ACM, New York, 174–183.

MYERS, B. A. 1985. The importance of percent-done progress indicators for computer-human interfaces. In *Human Factors in Computer Systems—CHI'85 Conference Proceedings.* ACM, New York, 11–17.

OSCHMAN, R. B., AND CHAPANIS, A. 1974. The effects of ten communication modes on the behavior of teams during co-operative problem solving. *Int. J. Man / Machine Syst. 6*, 579–619.

REDER, S., AND SCHWAB, R. G. 1990. The temporal structure of cooperative activity. In *Computer Supported Cooperative Work—CSCW'90 Conference Proceedings.* ACM, New York, 303–316.

RESNICK, P. 1992. HyperVoice: A phone-based CSCW platform. In *Computer Supported Cooperative Work—CSCW'92 Conference Proceedings.* ACM, New York, 218–225.

RESNICK, P., AND VIRZI, R. A. 1992. Skip and Scan: Cleaning up telephone interfaces. In *Human Factors in Computer Systems—CHI'92 Conference Proceedings.* ACM, New York, 419–426.

ROTHFEDER, J. 1992. *Privacy for Sale.* Simon and Schuster, New York.

RUTTER, D. R. 1987. *Communicating by Telephone.* Pergamon Press, New York.

SCHMANDT, C. 1993. Phoneshell: The telephone as computer terminal. In the *1st International Conference on Multimedia.* ACM, New York, 373–382.

SCHMANDT, C. 1990. Caltalk: A multi-media calendar. In *Proceedings of the 1990 Conference.* The American Voice I/O Society, San Jose, Calif., 71–75.

SCHMANDT, C. 1981. The Intelligent Ear: A graphical interface to digital audio. In *Proceedings of the IEEE Conference on Cybernetics and Society.* IEEE, New York, 393–397.

SCHMANDT, C., AND ARONS, B. 1985. Phone Slave: A graphical telecommunications interface. *Proc. Soc. Inf. Display 26*, 1, 79–82.

Soclof, M., and Zue, V. 1990. Collection and analysis of spontaneous and read corpora for spoken language system development. In *Proceedings of ICSLP.* 1105–1108.

Sproull, L., and Kiesler, S. 1991. *Connections: New Ways of Working in the Networked Organization.* MIT Press, Cambridge, Mass.

Stifelman, L. J. 1992. VoiceNotes: An application for a voice-controlled hand-held computer. Master's thesis, Massachusetts Institute of Technology, Cambridge, Mass

Stifelman, L. J. 1991. Not just another voice mail system. In *Proceedings of the 1991 Conference.* American Voice I/O Society, San Jose, Calif., 21–26.

Stifelman, L. J., Arons, B., Schmandt, C., and Hulteen, E. A 1993. VoiceNotes: A speech interface for a hand-held voice notetaker. In *Human Factors in Computer Systems—InterCHI'93 Conference Proceedings.* ACM, New York, 179–186.

Want, R., Hopper, A., Falcco, V., and Gibbons, J. 1992. The active badge location system. *ACM Trans. Office Inf. Syst. 10*, 1, 91–102

Watabe, K., Sakata, S., Maeno, K., Fukuoka, H., and Ohmori, T. 1991. Distributed desktop conferencing system with multiuser multimedia interface. *IEEE J. Sel. Areas Commun. 9*, 4, 531–539.

Weiser, M. 1991. The computer for the 21st century. *Sci. Am. 265*, 3 (Sept.), 66–75.

Wilcox, L., and Bush, M. 1991. HMM-based wordspotting for voice editing and indexing. In *Proceedings of Eurospeech 91.* 25–28.

Zellweger, P., Terry, D., and Swinehart, D. 1988. An overview of the Etherphone system and its applications. In *Proceedings of the 2nd IEEE Conference on Computer Workstations.* IEEE, New York, 160–168.

Zue, V. W. 1991. From signals to symbols to meaning. On machine understanding of spoken language. In *Proceedings of the 12th International Congress of Phonetic Sciences.*

Received January 1993; revised July 1993; accepted July 1993