Conference Scribe: Turning Conference Calls into Documents

Pierre D. Wellner*
AT&T Labs – Research
Red Bank, NJ 07701 USA

David M. Weimer
AT&T Labs - Research
100 Schulz Drive
P.O. Box 7033, Rm 3-201
Red Bank, NJ 07701 USA
dmw@research.att.com

Barry Arons**
AudioVelocity Inc.

* Currently at Spiderphone.com, Inc. pwellner@spiderphone.com

** Currently at Tellme Networks, Inc. barry@tellme.com

Abstract

Telephone calls and multi-party conference calls are transmitted and controlled by computer much like electronic documents. However, the way we traditionally use conference calls is very different from the way we use documents. This is primarily due to the transient unstructured nature of audio, and the lack of tools for managing, manipulating, and displaying telephone conversations and conference calls.

This work is aimed at letting people use conference calls in many of the same ways that they use documents. This paper describes a system for turning conference calls into archived documents that can be browsed, skimmed, displayed, hyperlinked and annotated on the World Wide Web. The "Conference Scribe" system has three principle components: the "SkimServer" for recording and playback, the "IDB Server" for storing and retrieving time stamped labels, and a Java applet for interactively skimming recordings. The parallels between conference calls and documents are reviewed, and form the basis for the design of the system.

1. Introduction

Written communication is normally created, transmitted, archived, displayed, and manipulated as documents. Voice communication, on the other hand, is typically not displayed or stored, despite the fact that digital speech is inexpensive to archive. For example, one year of telephone conversations can easily fit on a tape costing less than forty dollars if we assume four hours per day, 16kbs coding [2], and a DDS-3 DAT cartridge. So why not record all those phone calls? One reason most people don't record gigabytes of speech is that there are no flexible, efficient techniques to search through them, retrieve selected conversations, and then listen to them quickly. Despite these limitations, some calls are important enough to record anyway (e.g., multi-party conference calls). This paper describes a system for turning conference calls into archived documents that can be browsed, skimmed, searched, displayed, linked and annotated on the World Wide Web. The initial goal of this work to make recorded conference calls more functional and useful. If successful, a longer term goal may be to let all phone calls be turned into such valuable documents that some people will archive all their calls just as today some people archive all their e-mail.

1.1. Conference calls

Our initial focus is on multi-party conference calls over ordinary phone lines. The system can be used just as well for two-party calls, but we believe that conference calls benefit more from being turned into documents. The larger the conference, the more difficult it becomes to control and schedule, and the more likely that someone will need to review it later. These calls also tend to be long and thus more tedious to skim through.

Like most phone calls, conference calls are usually not recorded and exist primarily as a dynamic, ephemeral, and invisible form of communication. The lack of visible structure makes it difficult for people to discover call content afterward, and difficult to understand and control ongoing conferences as they take place. Participants are often not aware of who is speaking, who is listening, or who just dropped their connection. This attempt to turn conference calls into documents not only address persistent call recordings, but also presents the ongoing dynamic conference as a visible web document. The work described below includes two parts: the WebRooms system which provides a web-based graphical user

interface for live conference call status and control, and the Conference Scribe system which provides a graphical document interface to conference call recordings.

The current user interface for AT&T's teleconference service, is strictly through the telephone. To set up and control conference calls, customers speak with an operator. If any problems occur during the call (e.g., someone plays music-on-hold to the entire conference), an operator must be contacted before the problem can be resolved.

The WebRooms system described in this paper allows anyone with a web browser and telephone to control conference calls using a web-based graphical user interface. Audio is carried on ordinary phone lines, while telephone signaling, user interface events and graphics are sent over the web via HTTP protocol. The authors in collaboration with AT&T TeleConference Services developed the system, and used it in a limited access technical trial for two years.

Conference call recording is also available today from TeleConference Services. Customers dial in to hear playback of a conference at a specified time (a *rebroadcast*), or callers can dial in to individually hear recordings beginning at any time over the course of a few days. However, most customers don't bother to record or play back calls. Even people who want to hear certain parts of a conference are often unwilling to listen to the whole thing to find the small fragments that they are interested in. One exception, however, are what are referred to as "investor relations" calls. Large companies host these events to announce their quarterly results to Wall Street analysts. They typically consist of the CEO or CFO making a prepared statement followed by questions and answers with Wall Street analysts. Many analysts track a number of companies and cannot attend all conference calls, but they will take the time to listen to entire recordings before making their recommendations.

1.2. Conference Scribe

The Conference Scribe is a separate service that can record any WebRooms teleconference. When recording, the Conference Scribe appears just like any other participant on the conference call. It normally does not make any sound except a short initial beep to indicate that it has started recording. As it records the audio, it also stores automatically generated labels that identify specific intervals: when participants are added and dropped from the call, who talked when, and the location of any pauses in the speech. This labeled interval data is used to present users with a visual display of the conference recording and enable continuous real-time control of call playback speed and location from any Java-enabled web browser located near a telephone. The system currently plays the audio through the telephone rather than through the Internet so access does not require a special sound card or Internet connection. However, with the rapid deployment of audio streaming technology on the web, this may soon change.

2. Conference Calls as Documents

What does it mean to "turn conference calls into documents" when the term "document" has so many meanings? This paper first identifies five significant ways in which people use documents, then it describes a system that begins to allow similar use of phone calls. The subsections below detail five ways that people use documents that the system mimics for conference calls: *archiving, finding, displaying, linking,* and *labeling.* Although we routinely use electronic documents in these ways, we do not normally do the same with telephone conversations.

2.1. Archive

One key difference between documents and phone calls is that most documents are persistent: they can be stored somewhere and retrieved at a later time. Large collections of documents can be stored for later reference in a public, personal or group archive depending on access control mechanisms. They can also be copied and distributed for sharing with other people (e.g., through the web).

2.2. Find

People find documents in multiple overlapping ways. Sometimes we *browse* without a clear idea of exactly what we are looking for. At other times, a precisely directed *search* is performed with respect to a particular document property. Once a particular document is found, we often *skim* through it to find a specific part of interest.

2.3. Display

Documents are often designed and displayed with visual structure. Paragraphs, fonts, headings, tables, lines, and illustrations can be used to create visual structure, and we depend on the display for navigational aids to help us understand the structure of the content, and to skim through documents and find the parts we need. Phone calls are inherently non-visual, and so they are not usually displayed, but generating visual structure for them is an important part of turning them into documents.

2.4. Hyperlinks

A key feature of HTML documents on the web is that they can be linked *from* any other HTML document and

they can link *to* other web documents. To be fully part of the World Wide Web, documents that represent phone calls must be displayable from any browser and support URL hyperlinks both in and out.

2.5. Label

Most documents have intervals labeled with specific properties. In HTML documents, for example, parts of the text can be labeled with properties such as emphasized, heading, or a hyperlink. Although interspersed with the text, these labels or tags are distinct from the text, and they are used to control the display of a document. Document editors allow users to select regions of text and tag them with desired labels. Labels of this type are even more critical for voice documents because of the difficulty in presenting raw speech in ways that are visually meaningful. This work explores the use of both automatic and manual labeling.

3. Related Work

A number of systems have been developed that give phone calls or speech recordings some of the document properties discussed above. They include both research prototypes and commercial products. This section describes how these systems allow phone calls and voice recordings to be used like documents.

3.1. Voice logging recorders

Many call centers, financial traders, 911 dispatchers, security companies, prisons, etc., record and archive their telephone calls using systems known as "voice logging recorders." These systems are installed and used on premise for recording and playback of calls. Each call is tagged with information such as date, time, channel, duration, and phone number for use in searching specific calls. The systems do an excellent job of *archiving* calls. They are oriented towards relatively short phone calls, however, so they do not display call structure or help listeners skim through a particular conversation. Although some systems allow call recordings to be copied into PC audio files that can be emailed, etcetera [7], they are not designed to support web-based access and linking.

3.2. MIT Media Lab Speech Research Group

SpeechSkimmer [1] explored ways to play back recorded speech at multiple speeds and a within a hierarchy of "skimming levels" which played speech at several different levels of detail. These levels ranged from unprocessed full-length to very brief summaries of speech based on labels in the speech derived from energy and pitch information. SpeechSkimmer explicitly avoided the use of a graphical user interface to display speech, however, and instead focused on the design of a simple input device to control speech playback. It also did not attempt to archive phone calls, help find particular recordings, or link speech documents with web documents. It was, however, one of the inspirations for the Conference Scribe project.

The Listener [3] was a listening tool for the telephone that ran on a workstation. It allowed users to identify and save parts of a telephone conversation as it progressed. It used a microphone in the user's office to distinguish the local talker from remote talkers, and it displayed conversations on screen with alternating bars indicating who was talking. Captured telephone conversations were saved in local files and could be displayed and played back at any point.

3.3. CTI products

Certain computer telephony integration (CTI) products based, for example, on Microsoft's TAPI can control conference calls from special-purpose applications running on PCs with access to appropriate hardware and telephones either on the PC or its LAN. WebRooms differs from these products because it only requires a web browser and telephone. WebRooms call-control is no longer unique, however, because Sprint and others now offer similar web-based conference call control services.

3.4. Time-stamped note taking

A number of systems have been developed to record a stream of audio or audio/video and create a time-stamped index into the stream for the purpose of efficiently accessing it during playback. One early system was NoTime [5], which integrated a video camcorder with a portable pen-based notebook computer, allowing the user to sketch and take time synchronized notes on the LCD tablet while making a video recording. At *playtime*, the user could circle any mark on the tablet to play back the video from the point when that mark was made. Filochat [15] was a similar system that allowed hand written notes entered on an LCD tablet to augment audio recordings, and the Audio Notebook [12] provided similar functionality with the use of real paper.

These systems all demonstrated ways to label and link speech using graphical note-taking events. Visual representation of the speech itself was not significant because the primary interface for playback control were the handwritten graphical notes.

3.5. Xerox PARC

The PARC Etherphone project [13] pioneered many concepts of CTI and the use of digitized speech in electronic documents. It included an elegant server for storing and editing speech recordings as well as a viewer for displaying, editing, and playing them within electronic documents. Using custom-made telephones and the unique Cedar infrastructure, Etherphone in fact achieved many goals of our current work ten years earlier. Today, using the web, we can explore these ideas further in a new context where everyone with a telephone and web browser has access to the recording system.

Minneman *et al*. describe a "confederation of tools" for capturing, indexing, and accessing meeting activities for the purpose of meeting support both during a meeting and after [8][9]. The key application described in this work is use of a large white-board sized, pen-based computer (the LiveBoard) to draw figures and take notes during a recorded meeting, and to access the recording later, much like NoTime, Filochat, and the Audio Notebook. Unlike most of the other systems mentioned above, both the PARC work and Conference Scribe use a distributed object-oriented server-based architecture. Both are designed to allow multiple people independent use of the service at the same time. Conference Scribe is built using the telephone network. In contrast, the PARC work is built around specially instrumented meeting rooms and workstations.

A separate tool also developed at PARC displayed conversations along multiple "tracks" showing who was talking when, based on automatic speaker segmentation [4]. This tool demonstrated effective display and labeling of speech in a similar way to the Listener. The automatic segmentation was reported to work well with professionally recorded meetings with multiple microphones but not with telephone conference calls.

3.6. Pictorial Transcripts

Shahraray and Gibbon developed a system at AT&T Labs named Pictorial Transcripts which captures broadcast television news audio recordings along with key still-frames of the video and closed-caption text [11]. Search, retrieval, and playback of the audio and video stills is done over the web. A database of time stamped indices is stored along with the audio marking points where video stills were captured, and marking the time stamped words from the closed caption text. From the web, a user can use these indices to find the point in a recorded program when a specific topic was discussed, then listen to the audio from that point and see pictures that were shown at the same time.

On the surface, broadcast TV news may seem quite different from telephone conference calls, yet the Scribe project shares many common goals, problems, and techniques with Pictorial Transcripts because both deal with recorded speech on the web. Plans are underway for the two systems to share software. TV broadcast news includes closed-caption text and pictures, which we do not get with phone calls, yet the conference bridge outputs talker and connection data which we do not get with TV broadcast news. Scribe plays the audio over the phone while Pictorial Transcripts plays the audio over the Internet. Both applications would benefit from both kinds of playback, and eventually both should be able to share common interval database structures and tools for indexing, search, and retrieval of speech.

4. WebRooms

The WebRooms interface for conference setup and control was designed well before the Conference Scribe recording system. WebRooms is implemented with CGI programs that display conferences as HTML, while the Conference Scribe displays conference recordings using a Java applet. These implementation differences, along with the intent for both systems to be independently useful, led to two loosely coupled, but separate user interfaces.

4.1. WebRooms User Interface

The conference control interface was influenced by the functionality available from today's telephone-only user interface for AT&T TeleConference Services, and it can only be accessed after logging in as a registered user. To start a conference, the host enters people's names and numbers, or selects entries from a personal phonebook. The host can also restart a previously saved conference i.e., a "frequent meeting." Figure 1 shows how an active conference is displayed as a document.

The host controls each participant's phone line with buttons that dial, hang-up, mute, and delete. Other participants do not need to be on the web, but if they are and know the proper access code, they can join the conference and monitor it from a similar page with a restricted set of control buttons. Every view of an active conference is updated, via *server-push*, whenever the state

Figure 1. Active conference control.

of a connection changes. This allows the host and participants with web access to immediately see whose connection was added or dropped.

5. Conference Scribe User Interface

Recording a conference requires no change to the WebRooms user interface, because the "Conference Scribe" looks like any other participant to the user. This will change when some form of manual labeling is required during the recording process. For example, a set of controls might be added to the recorder connection to allow entering annotations.

The following subsections describe the conference playback document.

5.1 Use of labels to display structure

The conference playback document is implemented as a Java applet. It uses a visual structuring of the recording as a series of color-coded intervals plotted on a horizontal time axis in an area we call the *timeline window* (figure 2). Each participant in a call is allocated a separate timeline for graphically depicting all labeled intervals that are associated with that person (e.g., dialing, connected, muted, talking, etc.).[2] By plotting each interval type one at a time, starting with taller bars, the document displays overlapping intervals on the same line (figure 3). Intervals that are not associated with an individual person are plotted separately above the participants, (e.g., hyperlinks, speech segments, etc.). The timeline window provides a snapshot of every attendees participation, and can be used to navigate through the recording.

[2] An editing version of the UI allows intervals to be manually deleted; participants can be added and talker intervals can be created by clicking on a talker's name.

5.2. Controls for playing and skimming

Once users have established a phone connection to the player, they can use the tool bar below the timeline to begin playing the audio and adjust the skimming parameters.

The toolbar provides five buttons to control the player: "goto beginning", "jump back", "stop", "play", and "jump forward". It also contains a slider for adjusting the playing speed (0.7x, 1.0x, 1.3x, 1.7x, and 2.0x), a menu for selecting the zoom factor (none, 20min., 10min., and 5min.), and an on/off button for pause removal.

5.3. Using the timeline window

As the audio plays, a vertical red needle moves across the timeline. When the needle moves, every participant's nametag is colored to reflect that person's state at that time in the meeting. Figure 2 shows a one hour conference with the entire duration visible (zoom = none). In this view, the visual structures help make some details of the call immediately obvious. For example, the number and span of the light colored bars can identify the most/least dominant talkers. The initial long uninterrupted talking bands show who gave the formal presentations. Finally the point where the question and answer session began is visible roughly half way into the call, where many short talking intervals are scattered among many participants. More detailed information can be found by either listening to the audio or by searching through linked annotations, images, and other documents.

The zooming feature allows the user to change the duration displayed in the timeline window. A numbered scrollbar allows the user to register the zoomed-in portion with the full duration, and scroll using mouse clicks or arrow keys on the keyboard. Scrolling is independent of the player location needle, so the user can separately glance at regions, without disrupting listening. The player needle can be moved by clicking on the timeline, or by pressing a jump forward/backward button. When this happens the SkimServer plays a short non-speech audio cue and begins to play at the new location.

5.4. Hyperlinks

Clicking the timeline near the top is used to select hyperlinks rather than to move the needle. When a link is selected, or the "links" button is pressed, a dialog displays all the links in the recording. This dialog can be used to visit a link, edit a link, or create a link both in and out of the timeline. There are currently five types of link supported: *annotations*, *audio*, *documents*, *images*, and *general URL*. In fact all links are implemented using URLs except *annotations*, which store textual content as

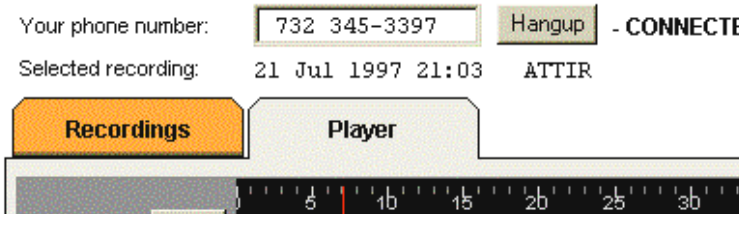

Figure 2. A conference playback document showing an investor relations call.

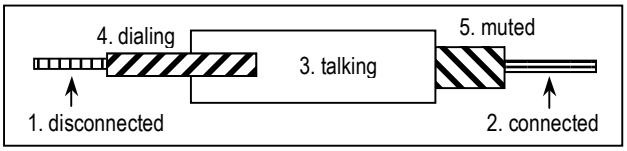

Figure 3. Detail of overlapping event intervals.

interval data. Each type is displayed on the timeline with a representative icon.

Hyperlinks into and out of the timeline are stored as intervals, and contain both a beginning and ending time offset. Thus a link can refer to a particular point or region of the timeline, allowing a rich set of skimming alternatives. For example, following a link can cause play to begin at a certain point, end at a certain point, or sequence through selected regions. This implies that following a link can have multiple effects, including moving the player needle and changing the document page. Exploiting such capabilities will hinge on making manual creation of such links easy for the user.

5.5. Display alternatives: timelines verses pagination

Our use of timelines is similar to the displays used in other work, where the call artifacts available were similar [3][4]. There are other alternatives that are less oriented toward the timeline, and more towards a sequence of key segments. The Pictorial Transcripts work mentioned earlier, relies on the consistent structure of broadcast television news, and uses images and segments of closed caption transcriptions to render pages resemble a searchable electronic news magazine. This approach seems well suited to a variety of situations where the source is highly structured, with visual material to augment the audio portions. The investor relations' conferences mentioned earlier, and depicted in figure 2, are one possible example. Blending these two approaches is accomplished to some degree through the use of hyperlinks.

6. Architecture

The main components of our system are the Conference Recorder, SkimServer, Interval Database (IDB) Server, and the Java user interface (see figure 4).

At record time, the conference host uses WebRooms to dial Conference Scribe as an additional participant to the conference. At the same time, a Conference Recorder process tells the IDB Server to create a new collection point (*depot*) for storing all data related to this particular recording, and it tells the SkimServer to begin recording an audio file using a Dialogic board.[3] While the conference is running, the Conference Bridge [6] detects call control events, which participant is talking, etc. and sends these events through WebRooms and the Conference Recorder into the new *depot*. Meanwhile, the SkimServer detects pauses in speech and adds these events as well.

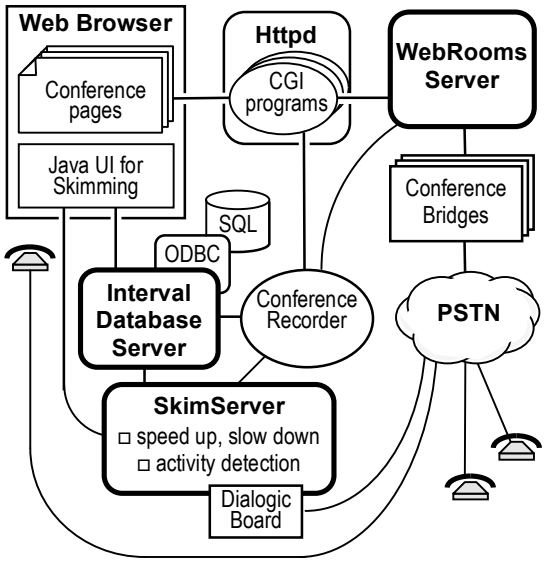

Figure 4. Architecture of WebRooms and Conference Scribe.

At *playtime*, the user brings up a Java user interface to select a recording accessed via the IDB server. The interface retrieves labeled interval data for this recording and uses them to display a visual timeline of events. The user enters a phone number that is passed to the SkimServer so it can call the user's telephone for conference playback through the Dialogic board. As the audio plays on the user's phone, the Java UI continuously updates the graphical display and controls how the recording is played using the SkimServer. Clients and servers communicate through CORBA application programming interfaces, making it easy for programs written in different languages running on different platforms to work together. WebRooms and the Conference Recorder are run on Sun Solaris platforms, while the Interval Database and SkimServer run on Windows NT.

6.1. SkimServer

The SkimServer performs the following functions:

1. Record audio from telephone line to file.
2. Detect speech events while recording and post them to the interval database.
3. Play from file to telephone line
   - from any point in recording
   - at variable speeds
   - with pauses removed or not

The SkimServer is based on the same type of hardware as standard voicemail servers, and it performs many of the same functions. The main difference between this server and a more traditional voicemail server is that it processes speech events and posts them to the IDB, and also that it provides fine control over what parts of the audio file are played and what parts are skipped. The code for speeding up and slowing down speech is based on [14].

6.2. Interval Database

Any system that supports browsing and visual display of archived speech needs to store and retrieve labeled interval data. This is data that describes properties about specific intervals within the speech, such as who is talking, pauses in speech, telephone call control data. This can be further extended to applications that require intervals that mark video scene changes, or relate automatic speech recognition output to a recording. At a minimum, each interval data element must include a reference to a specific speech recording, a start time, end time (or duration), and type-specific data. The labeled intervals can be created, stored, and retrieved by a number of different applications. Some are automatically derived from raw speech data, some are side effects of user activity, and others may be entered manually at record time or at play time.

6.2.1. Time synchronization

All applications that post events to the IDB must specify precise millisecond offsets for start and end times of each interval. All offsets are from an absolute start-time for the recording. Posting intervals from different machines in real-time requires all clients that are posting events have synchronized clocks, so standard NTP software is run on all of these machines.

[3] http://www.dialogic.com

6.2.2. Queries

Browse, search, and playback applications need to query and display subsets of interval data. Examples of queries that must be supported include:

- All interval data for a specific recording, sorted by time and type.
- All intervals of a specific type with specific values, or values within a particular range.
- Intervals within an absolute or relative time range.
- Intervals of a specific duration.

6.2.3. Logical/Set operations

Assume a user wants to see and/or hear only the parts of a recording when person `A` or person `B` was talking, and wants to leave all the pauses out. This might be expressed by making three queries: intervals when `A` was speaking `(set A)`, intervals when `B` was speaking `(set B)`, and pause intervals `(set P)`. One could express the desired set as `A union B less P`, or if we think of these sets as long bit masks, then we can describe them as logical operations: `(A | B) & (¬ P)`.

6.2.4. Probabilistic or fuzzy intervals

Some types of intervals may not have clear start or end times. Instead of a binary on/off state at each time increment, some data has an associated probability curve over time because the exact times of the events are not certain. For example, output from automatic speech recognition (e.g., phoneme lattices) can include several overlapping hypotheses about what words are being said at any given moment.

Initial implementations of the IDB do not include direct support for fuzzy intervals. It is possible, however, to use binary intervals along with a probability value in the type-specific numeric data field to achieve a similar effect, but without fuzzy logical operations.

6.2.5. Full text search on transcriptions

Transcriptions can be stored as interval data, perhaps one sentence per interval, or one word per interval depending on how fine a mapping is desired between words and time. The transcriptions may be produced from closed caption text, higher quality off-line transcriptions, or a lower quality automatic speech recognition system. We have not yet integrated our IDB with a full-text search engine.

6.2.6. Multi-user access control

Some recorded speech, such as broadcast news, is public. Access control for these kinds of recordings is relatively simple, perhaps only limiting access to users that have paid a subscription fee. Personal telephone conversations and voice mail can also have relatively simple access control requirements. A single user owns each recording and only that person has access to their recordings.

Things get more complicated, however, when people want to share certain recordings but not others. They also get more complicated for conference calls, where a specific set of people need access, but the access may need to be limited (e.g., only one person has permission to delete the recording). This problem is by no means unique to recorded audio archives. It occurs whenever files and services need to be shared among groups of people, and a variety of approaches have been developed to deal with this, including OS groups, Kerberos, and revision control systems. Today the IDB assigns a single owner to every recording, and that owner can assign an individual password to any recording. This allows anyone (including unidentified users) to access a recording if they know the password.

7. Future Work

We are exploring ways of enhancing feature extraction from the audio track of a call. This includes automatic speech recognition to support text-based searches, and speaker detection algorithms that can be used to extract speaker intervals when multiple participants share a single telephone line.

We also intend to explore a tighter coupling of live conference calls with their recordings. One goal would be to explore ways of marking intervals at record time through the conference control user interface. Another goal would be to allow a pause/resume feature to the conferencing service. Participants might join a conference call late, or need to step out for a moment. If a recording were in progress, the pause/resume feature would allow them the option of catching up gradually. This would work by playing what they missed at a faster rate, and seamlessly splicing it with the live call.

7.1. User Evaluation

Before we can begin evaluating the effectiveness of these tools, we have to address some of the unresolved issues.

Issue 1: What retrieval tasks, media, and techniques can we measure our tool against. *Investor relations*' calls are cases where phone calls are sometimes manually transcribed. An audio recording is all that is available most of the time, and any useful means of automated transcription is desired. Measurements of effectiveness may need to compare text-based search, audio-based search, and visual search.

Issue 2: How useful are fully automated feature-extraction and restructuring techniques, compared to manual structuring techniques. If manual techniques

outperform automated ones, what manual tasks are most people willing to do. Investor relations calls are an interesting case. For a short period of time their value is higher than other types of calls, and the added cost manual hyperlinking and restructuring may be justified. However, when this cost is too high, the typical investor relations call has a well-established format that lends itself to automated restructuring techniques.

Issue 3: How can the interface be made useful for larger groups of people. The example we showed in figure 2 had 17 speakers, but we are not sure this interface will scale well beyond 20 speakers. A related issue is what kinds of phone calls people will most want to record and review.

8. Conclusions

Much of what we have reported details the development of a client/server implementation of a service for archiving conference calls. Some of the elements in the service and in the user interface stems from related work in the field. Conference Scribe was developed to begin realizing in recorded phone-calls, some of the useful properties that we normally attribute to electronic documents. Our target for creating and accessing these archived phone calls is the World Wide Web, even though the voice is carried over the traditional telephone network.

We described how a web-controlled conference call can be recorded, and described the user interface for browsing and skimming a recording. Our goal is to eventually allow such efficiency in retrieving information from the audio, that recorded calls might be seen as in important office tool. The skimming interface we presented represents just one of many possible interfaces that we intend explore.

9. Acknowledgments

David Kapilow and Anthony Accardi provided us with the code for speeding up and slowing down the speech as well as the voice activity detector, which is based on unpublished work by David Malah. Greg Blonder, Al Milewski, and Carmel Smith gave us valuable comments and suggestions. David Gibbon and Chris Macey are helping us with new versions of the IDB Server.

10. References

[1] Arons, Barry *SpeechSkimmer: A System for Interactively Skimming Recorded Speech*. ACM Trans. on Computer Human Interaction. March 1997, Vol. 4, No. 1, pp. 3-38. www.media.mit.edu/~barons/tochi97.html

[2] Juin-Hwey Chen. Toll-Quality 16 kb/s CELP Speech Coding with Very Low Complexity, pp. 9-12, in Proc. of IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing (ICASSP), May 1995.

[3] Hindus, Debby, Schmandt, Chris, and Horner, Chris. Capturing, Structuring, and Representing Ubiquitous Audio. ACM Trans. on Information Systems, V11, N4, Oct. 1993.

[4] Kimber, Donald G., Wilcox, Lynn D. and Chen, Francine R. Speaker Segmentation for Browsing Recorded Audio, ACM CHI 95 pp 212-213 May 7-11 1995.

[5] Lamming, M.G. Towards a Human Memory Prosthesis. Tech Report #EPC-91-116. Rank Xerox EuroPARC, 1991.

[6] AT&T Digital Conferencing and Switching System (DCSS 96), Now manufactured by Lucent Technologies.

[7] Mercom Systems, Inc. *AudioLog: The Voice Logging Server with the Power, Reliability, and Flexibility of Windows NT*, http://www.mercom.com Lyndhurst, NJ.

[8] Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I., van Melle, B. A Confederation of Tools for Capturing and Accessing Collaborative Activity. In Proc. of ACM Multimedia 95.

[9] Moran, T., Chiu, P., Harrison, S., Kurtenbach, G., Minneman, S., van Melle, W. *Evolutionary Engagement in an Ongoing Collaborative Work Process: A Case Study*. In Proc. of CSCW, Boston, MA, Nov. 1996.

[10] Mowbray, T.J. and Zahavi, R. *The Essential CORBA: Systems Integration Using Distributed Objects*. John Wiley, New York, NY, 1995

[11] Shahraray, B. and Gibbon, D., *Automated Authoring of Hypermedia Documents of Video Programs*, Proc. Third Int. Conf. on Multimedia (ACM Multimedia 95), San Francisco, CA, November 1995.

[12] Stifelman, Lisa. Augmenting Real-World Objects: A Paper-Based Audio Notebook Proc. of CHI94, Vancouver, Canada. www.media.mit.edu/~lisa/anb.html

[13] Daniel Swinehart, Douglas Terry, and Polle Zellweger. *Etherphone: Collected Papers 1987-1988,* Xerox PARC Technical report CSL-89-2 May 1989 [P89-00002].

[14] Werner Verhelst and Marc Roelands, *An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech*, pp II-554-557 in Proc. of IEEE International Conf. on Acoustic Speech, and Signal Processing (ICASSP), April 1993.

[15] Whittaker, S. Hyland, P. and Wiley, M. Filochat: Handwritten notes provide access to recorded converstions. Proc. of CHI'94 271-277.