In *Proceedings of Hypertext (San Antonio, TX, Dec. 15–18),* ACM, New York, 1991, pp. 133–146.

Hyperspeech:
Navigating in Speech-Only Hypermedia

Barry Arons

MIT Media Laboratory
20 Ames Street, E15-353
Cambridge MA, 02139
E-mail: barons@media-lab.mit.edu

ABSTRACT

Most hypermedia systems emphasize the integration of graphics, images, video, and audio into a traditional hypertext framework. The hyperspeech system described in this paper, a speech-only hypermedia application, explores issues of navigation and system architecture in an audio environment without a visual display. The system under development uses speech recognition to maneuver in a database of digitally recorded speech segments; synthetic speech is used for control information and user feedback.

In this research prototype, recorded audio interviews were segmented by topic, and hypertext-style links were added to connect logically related comments and ideas. The software architecture is data driven, with all knowledge embedded in the links and nodes, allowing the software that traverses through the network to be straightforward and concise. Several user interfaces were prototyped, emphasizing different styles of speech interaction and feedback between the user and machine. In addition to the issues of navigation in a speech-only database, areas of continuing research include: dynamically extending the database, use of audio and voice cues to indicate landmarks, and the simultaneous presentation of multiple channels of speech information.

INTRODUCTION

Interactive "hypertext" systems have been proposed for nearly half a century [Bush45, Nels74], and realizable since the 1960's [Conk87, Enge68]. Attempts have continually been made to create "hypermedia" systems by integrating audio and video into traditional hypertext frameworks [Appl89, Back82]. Most of these systems are based on a graphical user interface paradigm using a mouse, or touch sensitive screen, to navigate through a two-dimensional space.

In contrast, the system described in this paper investigates a hyperspeech[1] application for presenting "speech as data," allowing a user to wander through a database of recorded speech without any visual cues. System architecture, conversational interfaces, and navigational aids for accessing information in a speech-only domain are also discussed.

While speech is a powerful communications medium, it exists only temporally—the ear cannot browse around a set of recordings the way the eye can scan a screen of text and images. Speech and audio interfaces must be sequential, while visual interfaces can be simultaneous [Gave86, Mull90]. These confounding features lead to significantly different design issues when using speech [Schm89], rather than text, video, or graphics.

[1] The word hyperspeech is used much like hypertext or hypermedia, as a generic term for speech-only hypermedia—it is not the name of the application described in this paper. A hyperspeech system is a subset of a more general class of hyperaudio systems.

Navigation in the audio domain is more difficult than in the spatial domain. Concepts such as highlighting, to-the-right-of, and menu selection, must be accomplished differently in audio than in visual interfaces. For instance, one cannot “click here” in the audio world to get more information—by the time a selection is made, time has passed, and “here” no longer exists.

While this paper focuses on the design and implementation of a particular hyperspeech application, it is intended to be a touchstone for a more general form of interaction with unstructured or semistructured speech data. Possible applications include the use of recorded speech, rather than text, as a brainstorming tool or personal memory aid. A hyperspeech system would allow a user to create, organize, sort, and filter “audio notes” under circumstances where a traditional graphical interface would not be practical (e.g., while driving) or appropriate (e.g., for someone who is visually impaired). Speech interfaces are particularly appealing in the context of small portable or handheld computers without keyboards or large displays.

RELATED WORK

Compared with traditional hypertext or multimedia systems, little work has been done in the area of interactive speech-only hypertext-like systems. Voice mail and telephone accessible databases can loosely be placed in this category, however they are far from what is considered “hypermedia.” These systems generally present only a single view of the underlying data, have a limited 12-button interface, do not encourage free-form exploration of the information, and do not allow personalization of how the information is presented.

The Rainbow Pages [Resn90] is a voice bulletin board service accessible through a telephone interface. The system is unique in that it encourages many-to-many communication by allowing users to dynamically add voice recordings to the database. The Rainbow Pages addresses issues of navigation among speech recordings, and includes commands such as “where am I” and “where can I go?”

Parunak [Paru89] describes five common hypertext navigational strategies in geographical terms. Several additional navigational aids that reduce the complexity of the hypertext database, “beaten path” mechanisms and typed links, are used in this application. A beaten path mechanism (e.g., a back-up stack or bookmarks) allows a user to easily return to places already visited.

Zellweger [Zell89] states “Users are less likely to feel disoriented or lost when they are following a pre-defined path rather than browsing freely, and the cognitive overhead is reduced because the path either makes or narrows their choices.” This hyperspeech application encourages free form browsing, allowing users to focus on accessing information rather than navigation. Zellweger’s paths are appropriate for scripted documents and narrations; this system focuses on conversational interactions.

Muller and Daniel’s description of the HyperPhone system [Mull90] provides a good overview of many important issues in voice-I/O hypermedia. They state that navigation tends to be modeled spatially in almost any interface, and that voice navigation is particularly difficult to map into the spatial domain. HyperPhone “voice documents” are a collection of extensively interconnected fine-grained hypermedia objects that can be accessed though a speech recognition interface. The nodes contain small fragments of ASCII text to be synthesized, and are connected by typed links. The hyperspeech system described in this paper differs from Hyperphone in that it is based on recordings of spontaneous speech rather than synthetic speech, there is no default path through the nodes, and no screen or keyboard interface of any form is provided.

SYSTEM DESCRIPTION

The Database

Audio interviews were conducted with five academic, research, and industrial experts in the user interface field[2]. Since only the oral content was of interest, all but one of the interviews was conducted by telephone. Note that videotaping similar interviews for a video hypermedia system would have been much more expensive, and difficult to schedule, than telephone interviews[3].

A short list of questions was discussed with the interviewees, helping them to formulate answers, before a scheduled telephone call. A telemarketing-style program then called, played recorded versions of the questions, and digitally recorded the response to each question in a different data file. Recordings were terminated using silence detection, without manual intervention. There were five short biographical questions (name, title, background, etc.), and three longer questions relating to the scope, present, and future of the human interface[4]. The interviews were deliberately kept short; the total time for each automated interview was roughly five minutes.

The interviews were manually translated into text with the assistance of a workstation-based transcription tool[5] [Aron91b]. The transcripts for each question were then categorized into major themes (summary nodes) with supporting comments (detail nodes). See Figure 1 for a schematic representation of the nodes in the database[6]. The starting and stopping points of the speech files corresponding to these categories were determined with a segmentation tool. Note that most of the boundaries between segments occurred at natural pauses between phrases, rather than between words within a phrase. This feature may be useful in future systems that automatically segment, and link, speech nodes.

[2] The interviewees were: Cecil Bloch (Somosomo Affiliates), Brenda Laurel (Telepresence Research), Marvin Minsky (MIT), Louis Weitzman (MCC Human Interface Group), and Laurie Vertelney (Apple Human Interface Group).

[3] One of the participants was in a bathtub during the telephone interview.

[4] 1) What is the scope, or boundaries, of the human interface? What does the human interface mean to you? 2) What do you perceive as the most important human interface research issues. 3) What is the future of the human interface? Will we ever achieve “the ultimate” human interface, and if so, what will it be?

[5] A serial mouse was built into a foot pedal; button events controlled the playback of the digital recordings.

[6] The node and link images are included here with some hesitation. Such images, intended only for the hypermedia author, can bias a user, or listener, of the system, forcing a particular spatial mapping onto the database. When the application is running, there is no visual display of any information.

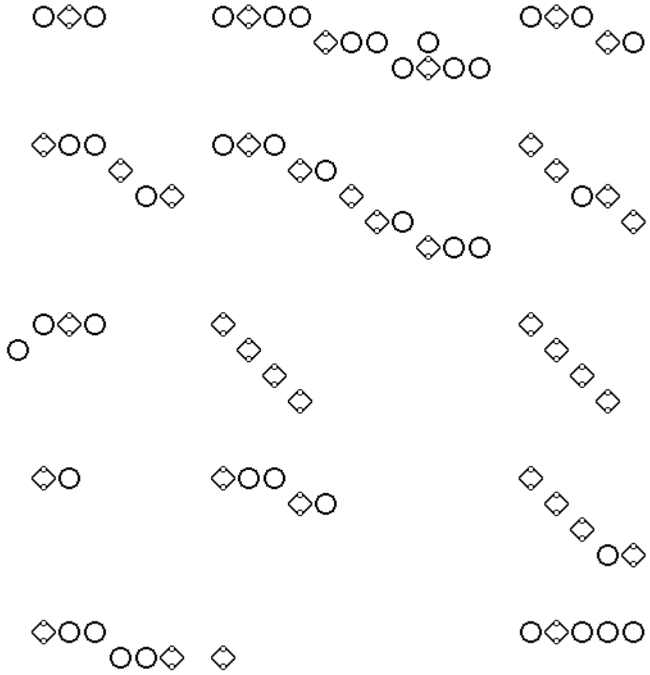

Figure 1: Graphical representation of nodes in the database. Horizontally contiguous nodes relate to the same summary node.

Of the data gathered[7] (approximately 19 minutes of speech, including trailing silences, um's, and pauses) over 70 percent was used in the final speech database. Each of the 80 nodes[8] contain short speech segments, with a mean length of 10 seconds (standard deviation of 6 seconds, maximum of 25 seconds). These brief segments parallel Muller's fine-grained hypermedia objects.

The Links

For this prototype implementation, an X Window System-based tool [Thom90] was used to link the nodes in the database. All the links in the system were typed according to function. Initially, a small number of *supporting* and *opposing* links between speakers was identified. For example, Minsky's comments about ". . . implanting electrodes and other devices that can pick information out of the brain and send information into the brain" are opposed to Bloch's related view that ends ". . . and that, frankly, makes my blood run cold."

[7] In the remainder of the paper, references to nodes and links do not include responses to the biographical questions.

[8] There are roughly equal numbers of summary nodes and detail nodes.

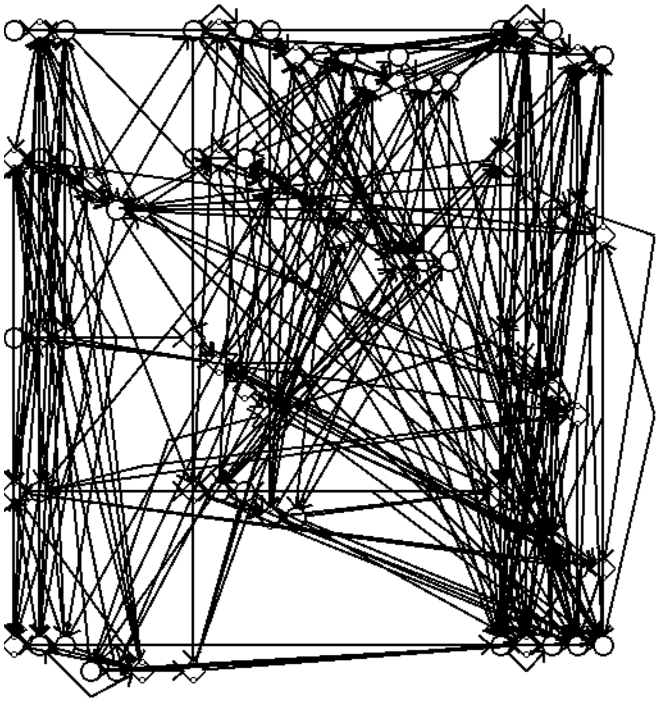

Figure 2: Graphical representation of all links in the database (version 2). Note that many links are overlaid upon one another.

As the system and user interface developed, a large number of links and new link types were added (there are 600 links in the current system). Figure 2 shows the links within the database[9]. The figure also illustrates a general problem of hypermedia systems—the possibility of getting lost within a web of links. The problems of representing and manipulating a hypermedia database become much more complex in the speech domain than with traditional media.

Hardware Platform

The telephone interviews were gathered on a Sun 386i workstation equipped with an analog telephone interface and digitization board. The hyperspeech system is implemented on a Sun SPARCstation, using its built-in codec for playing the recorded sound segments. The telephone quality speech files are stored uncompressed (8-bit mu-law coding, 8000 samples/second). A serial controlled text-to-speech synthesizer is used for user feedback. The recorded and synthesized speech sounds are played over a conventional loudspeaker system (Figure 3).

Isolated word, speaker dependent, speech recognition is provided by a board in a microcomputer; this machine is used as an RS-232 controlled recognition server by the host workstation [Aron89, Schm88]. A headset-mounted noise canceling microphone provides the best possible recognition performance in this noisy environment with multiple sound sources (user + recordings + synthesizer).

[9] A more appropriate authoring tool would provide a better layout of the links and visual differentiation of the link types.

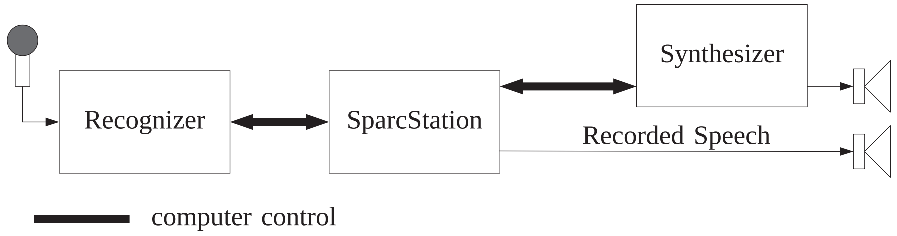

Figure 3: Hardware configuration.

Software Architecture

The software is written in C, and is running in a standard Unix operating environment. A simple stack model tracks all nodes that have been visited, and permits the user to return (pop) to a previously heard node at any time.

Because so much semantic and navigational information is embedded in the links, the software that traverses through the nodes in the database is straightforward and concise. This data driven architecture allows the program that handles all navigation, user interaction, and feedback to be handled by approximately 300 lines of C code[10]. Note that this data driven approach allows the size of the database to scale up without having to modify the underlying software system.

USER INTERFACE DESIGN

The user interface evolved during development of the system—many improvements were made throughout an iterative design process. Some of the issues described in the following sections illustrate the differences between visual and voice interfaces, and are important design considerations for implementors of similar speech-based systems.

Version 1

The initial system was sparsely populated with links and had a simple user interface paradigm: explicit menu control. After the sound segment associate with a node was played, a list of valid command options (links to follow) was synthesized. The user then spoke her selection, and the cycle was repeated.

The initial system tried to be “smart” about transitioning between the nodes. After hearing the recording, if there were no links that exited that node, the system popped the user back to the previous node, as no valid commands could be issued (i.e., links to follow). This automatic return-to-previous-node was potentially several levels deep. Also, once a node had been heard, it was not mentioned in succeeding menu prompts to keep the prompts as short as possible since it was assumed that a user would not want to be reminded about the same node twice.

Navigation in this version was difficult. The user was inundated with feedback from the system—the content of the recordings became lost in the noise of the long and repetitive menu prompts. The supposedly “smart” node transitions and deletion of menu items brought the user to unknown places, and left her stranded without landmarks because the menus were constantly changing.

[10] This number does not include extensive library routines and drivers that control the speech I/O devices.

Version 2

This section describes the current implementation of the hyperspeech system. The most significant change from the previous prototype was that a large number of links, and variety of new link types, were added.

A *name* link will transition to a node of a particular speaker, for example, user input of *Minsky* causes a related comment by Marvin Minsky to be played. Links were also added for exploring the database at two levels of detail. The *more* link allows a user to step through the database at the lowest level of detail. The *browse* link permits a user to skip ahead to the next summary node without hearing the detailed statements. The *scan* command automatically jumps between the dozen or so nodes that provide a high-level overview path through the entire database; this is similar to the scan feature found on many digital radios.

In order to reduce the amount of feedback to the user, the number of links was greatly increased so that a link of every type exists for each node. Since any link type can be followed from any node, command choices are uniform across the database, and menus are no longer needed. This is analogous to having the same graphical menu active at every node in a hypermedia interface; clicking anywhere produces a reasonable response, without having to explicitly highlight active words or screen areas. The table below shows the current vocabulary and link types of the system.

Link type	Command	Description
name	*Bloch* *Laurel* *Vertelney* *Weitzman* *Minsky*	Transition to related comments from a particular speaker
dialogical	*supporting*	Transition to a node that supports this viewpoint
	opposing	Transition to a node that opposes this viewpoint
control	*more*	Transition to next detail node
	continue	Transition to next detail node (alias for *more*)
	browse	Transition to next summary node
	scan	Play path through selected summary nodes
	return	Pop to previous node
	repeat	Replay current node from beginning
help	*help*	Synthesize a brief description of current location
	options	List current valid commands
on/off	*pay attention*	Turn on speech recognizer
	stop listening	Turn off speech recognizer

A host of minor changes made the system more interactive and conversational. In voice systems, time, not screen real estate, is the most valuable commodity. The speech segments in the database are, by default, played back 1.25 times faster than they were recorded without a change of pitch [Foul69, Maxe80]. If the *repeat* command is invoked, the node is replayed at normal speed for maximum intelligibility. The speaking rate of the synthetic speech has also been significantly increased to reduce user feedback time. Short repetitive types of feedback (e.g., direct echoing of recognized commands) are spoken at a faster rate than help or navigation-related feedback. The output volume levels have also been adjusted so that the primary output of the system, the recordings, is louder than the synthetic speech.

A sample interactive dialog with the current implementation of the system sounds like this:

Speaker	Utterance	Comments
Minsky	What I think will happen over the next fifty years is we'll learn more and more about implanting electrodes, and other devices, that can pick information out of the brain and send information into the brain.	
User	*opposing*	User objects to idea, does any one else?
Bloch	The ultimate is obviously some sort of direct coupling between the nervous system and artificial devices, and that, frankly makes my blood run cold.	
User	*browse*	Go to next summary from Bloch.
Bloch	In terms of ultimate development, I think that the thing that can be said is that it is unpredictable.	
User	*weitzman*	What is Weitzman's view?
Weitzman	I would hope that we never do achieve the ultimate interface.	
User	*continue*	Get more information.
Weitzman	We'll always be able to improve on it, and just the fact that during the process of getting there. . .	
User	*help*	Interrupt to get information.
Synthesizer	This is Louie Weitzman on the future of the human interface.	
Weitzman	. . . we are going to learn new things and be able to see even better ways to attack the problem.	Continue playing comment.
User	*Vertelney*	What does the industrial designer think?
Vertelney	I think it like back in the Renaissance. . .	
User	*return*	Not of interest. Interrupt, and go back to previous node.
Weitzman	We'll always be able to. . .	Weitzman again.
User	*Minsky*	What's Minsky's view of the future?
Minsky	And when it becomes smart enough we won't need the person anymore, and the interface problem will disappear.	

By default, explicit echoing [Haye83] of recognized commands is no longer used[11]. As soon as a spoken command is recognized, speech output (synthesized or recorded) is immediately halted, providing crucial feedback to the user that a command was heard. The system response time is fast enough that a rejection error[12] is immediately noticeable. Observers and first time users of the system are often more comfortable with the interface if command echoing is turned on. If a substitution error[13] occurs, the user can quickly engage the machine in a repair dialog [Schm86]. Note that speech recognition parameters

[11] At startup time the system can be configured for various degrees of user feedback.

[12] Rejection error: a word was spoken, but none was recognized.

[13] Substitution error: a word was spoken, but a different word was recognized.

are typically set so that substitution errors are less common than rejection errors. A repair dialog (with command echoing) might sound like:

Speaker	Utterance	Description of action
User	*Weitzman*	Desired command is spoken
Synthesizer	*supporting*	Echoing (substitution error)
Minsky	"The interfa . . ."	Incorrect sound is started
User	*return*	Interrupt recording, pop to previous node
User	*Weitzman*	Repeat of misrecognized command
Synthesizer	"Weitzman"	Echo of correctly recognized word
Weitzman	"I hope we never do achieve the ultimate interface . . ."	Desired action is taken

LESSONS LEARNED ABOUT NAVIGATING IN SPEECH SPACE

Einstein is reported to have once said, "make everything as simple as possible, but not too simple." This idea also holds true in user interfaces, particularly those involving speech. Since time is so valuable in a speech applications, every effort must be made to streamline the interactions. However, if things are made too simple, the interface also can fall apart because of the lack of identifiable landmarks. Keeping the feedback concise, or allowing various degrees of feedback to be selected helps keep the interaction smooth and efficient. Grice's four maxims[14] about what, and how, something is said [Gric75], are perhaps more applicable in machine-to-human dialogs then they are in human-to-human conversations. These maxims capture many of the key ideas necessary for streamlining conversational interfaces.

The design of this system is based on allowing the user actively drive through the database rather than being passively chauffeured around by menus and prompts. This ability is based, in part, on having a fixed set of navigation commands that are location independent—from any location in the database, any command can be used[15] (i.e., any link type can be followed). In order to make the interactions fluent, transitions from one interaction mode to another (e.g., recognition to playback) must be designed for low system response time [Aron89, Schm89]. Similarly, any action by the system must be easily interruptible by the user. The system should provide immediate feedback to the user that an interrupt was received; this usually takes the form of instantly halting any speech output, then executing the new command.

One general advantage of speech over other types of input modalities is that it is goal directed. A speech interface is uncluttered with artifacts of the interaction, such as menus or dialog boxes. The recognition vocabulary space is usually flat and always accessible. This is akin to having one large pull-down menu that is always active, and contains all possible commands.

Authoring is often the most difficult part of hypermedia systems; hyperspeech systems have the added complication of the serial and non-visual nature of the speech signal. Recorded speech cannot be manipulated on a display in the same manner as text or video images. Note that schematic representations of speech signals (or transcriptions) can be viewed in parallel and handled graphically, but that the speech segments represented still cannot be heard simultaneously. The most efficient way to manually author a hyperspeech database today is though the use of transcriptions.

[14] Summary of points of interest: be as informative as required, be relevant, avoid ambiguity, and be brief.

[15] Note that this scheme may be difficult to implement in systems with a much larger number of nodes or link types.

PLANS FOR FUTURE VERSIONS

The hyperspeech application presented raises as many questions as it answers. There are many improvements and extensions that can be made in terms of basic functionality and user interface design. Some of the techniques proposed in this section are intriguing, and are presented to show the untapped power of the speech communication channel.

Command Extensions

There are a variety of extensions planned in the area of user control and feedback. Because of the difficulty of creating and locating stable landmarks in the speech domain, it is desirable to be able to add dynamically personalized bookmarks. While listening to a particular sound segment the user may say "*bookmark:* handheld computers," creating a new method of accessing that particular node. Note that the name of the bookmark does not have to be recognized by the computer the first time it is used. Instead, after recognizing the key phrase *bookmark*, a new recognizer template is trained on-the-fly with the utterance following the key phrase (i.e., "handheld computers"). A subsequent "*goto:* handheld computers" command, will take the user back to the appropriate node and sound segment.

Besides adding links, it is desirable to extend the database dynamically by adding new nodes. For example, using a scheme similar to that of adding bookmarks, the user can say[16] "*add supporting:* conversational interfaces will be the most important development in the next 20 years." This will create new *supporting* and *name* links, as well as a node representing the newly recorded speech segment[17]. A final variant of this technique is to dynamically generate new link types. For example, a command of the form "*link to:* handheld computers, *call it:* product idea" would create a *product idea* link between the bookmark and the currently active node[18].

There are many voice commands that can be added to allow easier navigation and browsing of the speech data. For example, a command of the form "*Laurel on Research*" would jump to a particular speaker's comments on a given topic. It is also possible to add commands, or command modifiers, that allow automated cross-sectional views or summary paths through the database. Command such as "*autoscan Minsky*" or "*autoscan future*" would play all of Minsky's comments or all comments about the future of the human interface. It may also be possible to generate on-the-fly arguments between the interviewees. A command such as "*contrast Minsky and Vertelney on the scope of the human interface*" could create a path through the database simulating a debate.

Audio Effects

Audio cues can provide an indication of the length of a given utterance, a feature particularly useful if there is a wide range of recording lengths. Some voice mail systems for example, inform the user "this is a long message" before playing a long-winded recording[19] [Stif91]. In a hyperspeech application, where playing sounds is the fundamental task of the system, a more efficient (less verbose) form of length indication

[16] An isolated word recognizer can be trained with short utterances (e.g., "add supporting") in addition to single words. Some of the examples presented in this section, however, would be better handled by a continuous speech recognizer.

[17] One complication of this design is that it may create nodes that are under populated with links. This may not present a problem if such nodes are sparsely distributed throughout the database.

[18] Many links can be generated, including *product idea* and *name* links in both directions.

[19] Note that it is counterproductive to say "this is a short message."

is desired. For example, playing a short (approximately 50 milliseconds) high pitched tone might indicate a brief recording, while a long (perhaps 250 ms) low tone suggests a lengthy recording [Bly82, Buxt91].

Ideally, a system designer would like to present multiple sound streams simultaneously without overloading the user’s cognitive abilities. Such simultaneity can help in solving the time congestion problem in audio interfaces [Aron91, Cher53]. While the “cocktail party effect”—the ability to focus one’s listening attention on a single talker among a cacophony of conversations and background noise—is often alluded to, few user interfaces have been built that exploit this audio stream segregation ability of humans [Breg90]. It may possible to enhance the primary speech signal so that it “remains in auditory focus,” compared with secondary or background channels. In this scheme, the speech signals are identifiable and differentiable, so that the user can shift her attention between the various sound streams [Ludw90, Cohe91]. This allows for a different form of speech navigation—the ability to move between “overheard” conversations.

In addition to cues that suggest the absolute length of a speech segment, it may be useful to provide hints that a sound is about to end while it is being played. In spoken English, sentences almost universally end with a “final lowering” of pitch. A hyperspeech system could exaggerate the impending end of a sentence by artificially lowering the pitch of the recording. This could provide the user with an indication that a decision or branch point is approaching. Doppler effect frequency shifts of a speech segment can also suggest that the user is approaching, or passing, a hyperspeech branch that exists only in time.

A final technology that may help untangle the web created by a hyperspeech application is a spatial audio system that can synthetically place sound source locations in space [Wenz88, Wigh30]. Such a system could add a strong spatial metaphor to a hypermedia interface [McKe91], either by explicitly placing the individual voices in particular spatial locations, or allowing the user to create a personalized map of the conversations in space.

Authoring tools

The program that gathered the interviews asked a series of questions that serve as the foundation of the organization of the hyperspeech database. In this application the questions were very broad, and much manual work was required to segment and link all the nodes in the database. If specific questions are asked, it is possible to segment and organize recorded messages automatically [Schm84]. With short segments of known semantic content, it is straightforward to automatically generate a large number of links [Aron91a].

CONCLUSIONS

The hyperspeech system described in this paper provides an introduction to the possibilities of constructing speech-only hypermedia environments. An implication of hyperspeech is that it is significantly different to create and navigate in speech-only hypermedia than it is to augment, or use, visually-based hypermedia with speech.

The system is unique in that it presents multiple views on a speech database; there are variety of ways to get from one specific node to another. The interface is goal directed; speech input provides a form of direct addressing that is difficult to capture in other interfaces—the user feels that she is navigating and in control.

The system is compelling to use, or listen to, for many reasons. Interacting with a computer by voice is very powerful, particularly when the same modality is used for both input and output. Speech is a very rich communications medium, layers of meaning can be embedded in intonation that cannot be adequately captured by text alone. Listening to speech is “easier” than reading text—it takes less effort to listen to a lecture than to read a paper on the same subject. Finally, it is not desirable, or necessary, to look at a screen during an interaction. The bulk of the user interface of this system was debugged by conversing with the system while wandering around the room and looking out the window. In speech-based systems, the hands, eyes, and body are free.

It is difficult to capture the interactive conversational aspects of the system by reading a written description. Most people who have heard this interface have found it striking, and its implications far reaching. One user of the system felt that she was “creating artificial conversations between Minsky and Laurel” and that the ability to stage such conversations was very powerful.

There are many technological, user interface, and social issues that must be solved in order for hyperspeech systems to be accepted as a new interactive medium. Techniques and methodologies for navigating in the speech domain are immature and unexplored, as are applications that are appropriately matched to such an interface. While this paper has focused on details of a particular system, it is perhaps most useful in terms of the questions it raises for future speech interfaces and hypermedia systems.

ACKNOWLEDGEMENTS

Many thanks to the user interface experts (see footnote 2) that provided the great speech material to work with. Marc Davis, Chris Schmandt, and Lisa Stifelman provided valuable input while the system was being designed, and in the organization and content of this paper. Comments from the anonymous reviewers were helpful and encouraging. Kris Thorisson helped collect a portion of the speech data, and created a linear presentation (with added video images) based on the same interview material.

This work was sponsored by Apple Computer and Sun Microsystems.

REFERENCES

[Appl89] Apple Multimedia Lab. The visual almanac: An interactive multimedia kit, 1989.

[Aron89] B. Arons, C. Binding, K. Lantz, and C. Schmandt. The VOX audio server. In *Proceedings of the 2nd IEEE ComSoc International Multimedia Communications Workshop*. IEEE Communications Society, April 1989.

[Aron91a] B. Arons. Authoring and transcription tools for speech-based hypermedia systems. In *Proceedings of 1991 Conference* American Voice I/O Society, September 1991.

[Aron91b] B. Arons. The cocktail party effect: Can it be exploited in speech communication systems? Submitted to Avios Journal, 1991.

[Back82] D. S. Backer and S. Gano. Dynamically alterable videodisc displays. In *Proceedings of Graphics Interface 82*, 1982.

[Bly82] S. Bly. Presenting information in sound. In *Proceedings of the CHI '82 Conference on Human Factors in Computer Systems*, pages 371-375. ACM, 1982.

[Breg90] A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[Bush45] V. Bush. As we may think. *Atlantic Monthly*, 176(1):101-108, July 1945.

[Buxt91] W. Buxton, B. Gaver, and S. Bly. *The Use of Non-Speech Audio at the Interface*. ACM SIGGCHI, 1990. Tutorial Notes.

[Cher53] E. C. Cherry. Some experiments on the recognition of speech, with one and two ears. *Journal of the Acoustic Society of America*, 25:975-979, 1953.

[Cohe91] M. Cohen and L. F. Ludwig. Multidimensional window management. *International Journal of Man/Machine Systems*, 34:319-336, 1991.

[Conk87] J. Conklin. Hypertext: an introduction an survey. *IEEE Computer*, 20(9):17-41, September 1987.

[Enge68] D. C. Engelbart. Presentation for the Fall Joint Computer Conference, 1968.

[Foul69] W. Foulke and T. G. Sticht. Review of research on the intelligibility and comprehension of accelerated speech. *Psychyological Bulletin*, 72:50-62, 1969.

[Gave89] W. W. Gaver. Auditory icons: Using sound in computer interfaces. *Human-Computer Interaction*, 2:167-177, 89.

[Gric75] H. P. Grice. Logic and conversation. In Cole and Morgan, editors, *Syntax and Semantics: Speech Acts*, volume 3, pages 41-58. Academic Press, 1975.

[Haye83] P. J. Hayes and R. Reddy. Steps towards graceful interaction in spoken and written man-machine communication. *International Journal of Man/Machine Systems*, 19:231-284, 1983.

[Ludw90] L. Ludwig, N. Pincever, and M. Cohen. Extending the notion of a window system to audio. *IEEE Computer*, 23(8):66-72, August 1990.

[Mexe80] N. Maxemchuk. An experimental speech storage and editing facility. *Bell System Technical Journal*, pages 1383-1395, October 1980.

[McKe91] D. McKerlie and M. Stevens-Guille. Quadra-phonic hyperspace interface. Personal communication. Proposal for a spatial sound system for auditory navigation of hypertext documents. HCI Design Laboratory, University of Guelph, 1991.

[Mull90] M. J. Muller and J. E. Daniel. Toward a definition of voice documents. In *Proceedings of COIS '90*, 1990.

[Nels74] T. Nelson. *Computer Lib: You can and must understand computers now*. Hugo's Book Service, 1974.

[Paru89] H. V. D. Parunak. Hypermedia topologies and user navigation. In *Hypertext '89 Proceedings*, pages 43-50. ACM, 1989.

[Resn90] P. Resnick and M. King. The rainbow pages: Building community with voice technology. In *Proceedings of DIAC-90, Directions and Implications of Advanced Computing*, Boston, MA, July 1990.

[Schm84] C. Schmandt and B. Arons. A conversational telephone messaging system. *IEEE Transactions on Consumer Electronics*, CE-30(3):xxi-xxiv, August 1984.

[Schm86] C. Schmandt and B. Arons. A robust parser and dialog generator for a conversational office system. In *Proceedings of 1986 Conference*, pages 355-365. American Voice I/O Society, 1986.

[Schm88] C. Schmandt and M. McKenna. An audio and telephone server for multi-media workstations. In *Proceedings of the 2nd IEEE Conference on Computer Workstations*, pages 150-160. IEEE Computer Society, March 1988.

[Schm89] C. Schmandt and B. Arons. Desktop audio. *Unix Review*, October 1989.

[Stif91] L. J. Stifelman. Not just another voice mail system. In *Proceedings of 1991 Conference*. American Voice I/O Society, 1991.

[Thom90] G. S. Thomas. *Xsim 2.0 Configurer's Guide*, 1990. Xsim, a general purpose tool for manipulating directed graphs, particularly Petri nets, is available from cs.washington.edu. by anonymous ftp.

[Wenz88] E. M. Wenzel, F. L. Wightman, and S. H. Foster. A virtual display system for conveying three-dimensional acoustic information. In *Proceedings of the Human Factors Society 32nd Annual Meeting*, pages 86-90, 1988.

[Wigh89] F. L. Wightman and D. J. Kistler. Headphone simulation of free-field listening I: Stimulus synthesis. *Journal of the Acoustic Society of America*, 85:858-867, 1989.

[Zell89] P. T. Zellweger. Scripted documents: A hypermedia path mechanism. In *Hypertext '89 Proceedings*, pages 1-14. ACM, 1989.