Observations on Using Speech Input for Window Navigation

Chris Schmandt, Debby Hindus, Mark S. Ackerman and Sanjay Manandhar

Media Laboratory, M.I.T.
Cambridge, MA 02139 USA
email:geek@media-lab.media.mit.edu

We discuss the suitability of speech recognition for navigating within a window system and we describe *Xspeak*, an implementation of voice control for the X Window System. We made this interface available to a number of student programmers, and compared the use of speech and a pointer for window navigation through empirical and observational means. Our experience indicates that speech was attractive for some users, and we comment on their activities and recognition accuracy. These observations reveal pitfalls and advantages of using speech input in windows systems.

Introduction

Considering the high expectations of speech input technology, there have been few convincing studies of its utility in an office environment. This paper describes an evaluation of speech recognition in a computer window system, where speech may provide an auxiliary channel to support window navigation tasks. In this study, speech was seen as assuming some of the functions currently assigned to the mouse, rather than as a keyboard substitute. We expected that allowing users to keep their visual and manual attention on the keyboard and the screen could provide an improved interface.

To do this, we built *Xspeak*, a speech interface to the X Window System. Xspeak allows words to be associated with each window; a window rises to the front of the screen and the cursor moves into it when the window's name is spoken. Thus a number of windows can be managed without removing hands from the keyboard or eyes from the screen.

We evaluated this interface empirically to determine the tradeoffs between voice and mouse navigation. However, we were not looking to compare the relative merits of voice input versus mouse input. Voice is an additional input medium that may augment the mouse in some respects and supplant it in others; some pointer operations, such as moving a window, would be difficult with voice alone.

We also wished to observe the acceptance and utility of this voice interface. A group of student programmers used Xspeak for several months. We were interested in whether these subjects would choose voice and under what circumstances, and how the addition of voice input would change their window system use.

In the next section we discuss issues in speech recognition as a user interface. The section following addresses navigation in window systems and the role voice might play. Following these, we describe our methodology and results.

Speech Recognition

Although speech recognition has received much positive publicity, the actual devices available today leave much to be desired, particularly in terms of recognition accuracy. Many variables affect error rates, including vocabulary size and composition, user's attitudes and speaking style, ambient noise, and microphone type and placement [Nusbaum, 1986, Biermann, 1985]. In short, it is difficult to get recognition to work well outside of controlled laboratory conditions. Because of these difficulties, the most successful applications for recognition to date have been in hands-and-eyes-busy situations [Visick, 1984], e.g., baggage sorting [Nye, 1982] or inspections of printed circuit boards [Harper, 1985], where the user is visually connected to the instrument. These are cases where the added benefit of hands-free input may outweigh other device-related problems.

The role of speech recognition in the office has yet to be established. There is little conclusive evidence that recognition is superior to the keyboard for data entry, much less for free-form typing and editing. For an excellent survey of the literature, we refer the reader to Martin [Martin, 1989]. Voice input may be more valuable when used in *conjunction* with other input devices (such as keyboard and mouse) for situations in

*This work was funded by the MIT X Consortium and Sun Microsystems.

in Interact 1990 Conference Proceedings.

which different tasks may be multiplexed across the different input modalities. To the extent that the tasks are separable, a performance improvement may be expected by splitting the input [Wickens, 1981].

Such considerations led Martin to design an experiment using speech recognition as an alternate input channel in a CAD system employing both keyboard and mouse. Her subjects were indeed more productive with the addition of voice, which she attributed in part to the speed of speech recognition versus typing longer command names, and in part to the ability of users to split attention across channels; that is, to remain visually focused on the screen while using speech. This second finding was particularly interesting in terms of expected utility of speech as an interface to a potentially visually complex window system. This paper was pivotal in motivating us to build our speech interface.

Window Navigation and Voice

Window systems allow the screen to be divided into smaller regions of input and output. Windows are used to organize work spatially, and, to a lesser extent, to perform tasks in parallel. For example, a user may have one window in a semi-permanent location on the screen running a text editor, and another window for the debugger.

There has been surprisingly little study of how people use windows, in terms of number, degree of overlap, distribution of tasks, or reasons for preference of a particular window system interface. Gaylin [Gaylin, 1986] discusses frequency of use of some window operations. A key study by Bly [Bly, 1986] compared tiled and overlapped windows in a task involving searching for information between windows. When the text to be searched was not all visible (in a tiled situation), she found that overlapping windows were more effective, with an interesting bimodality. For the most experienced users, overlapping windows were faster. For some less experienced users, overlapping windows were significantly slower. She attributed this to the added navigational tasks of manipulating the various windows. Overlapping windows were preferred among her users despite this added load.

This suggested to us that in a complex window environment, especially with users who like to create a large number of windows, an interface designed to improve navigation might be beneficial, providing faster access to various windows. Further, to the extent that navigation might be differentiated as a separate task from the activities occurring within each window, multi-modal input might lessen the user's cognitive load. This might allow successful use of a larger number of windows dedicated to specific tasks.

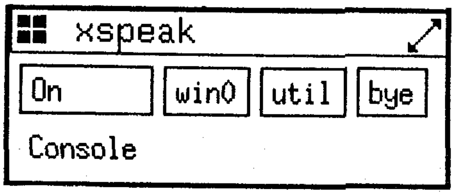

Figure 1: Xspeak control panel

Our application, Xspeak, allows access to windows by voice in the X Window System. Speaking a window's name causes it to pop up to the foreground and moves the mouse pointer to the middle of the window, at which point the window receives keyboard input focus. Thus users may move between windows and rearrange them without removing their hands from the keyboard.

Xspeak includes a graphical control panel (see Figure 1) which provides additional feedback on recognition results and can invoke utility functions to test, calibrate, and retrain the recognizer. (See [Schmandt, 1990] for a more detailed description of Xspeak operations.)

Xspeak runs on Sun workstations (it should run on any X server) using a Texas Instruments speech card in a PC-based audio server. We mounted a super-cardioid microphone (Sennheiser ME-80) on a stand next to the workstation screen, pointing out and at the user.

A consequence of not using noise canceling microphones is the tendency to pick up background noise as speech, that is, *insertion* errors. Recognizers are in general poor at discriminating whether a particular word is within their universe of names, being optimized to determine *which* known word was spoken. Since the consequence of insertion errors is window reconfiguration, which can be especially annoying if keyboard noise caused the error (suddenly user input goes to the wrong window), we set a high rejection threshold on the recognizer.

Evaluation Methodology

Having built this speech interface, we wanted to find out how it would be used, how it would affect users' overall workstation usage, and what problems existed with the interface that we needed to address. Since there are few other examples of speech interfaces to desktop windowing environments, we initially wanted to collect observations of this new interface in use. At this early stage of discovery, observation was more suited to our task of hypothesis generation.

Much of the work done on speech interfaces has simulated the recognition hardware [Gould, 1978]. This is due in part to the considerable practical problems of making speech recognition technology work reliably in relatively uncontrolled environments. However, we felt that it was important to observe usage over a longer period of time than is possible with simulations. In particular, how would having speech affect other post-acclimatization interactions? We preferred that our users be doing real work, since the artificiality of assigned tasks could confound our attempts to understand the ramifications of speech.

Furthermore, we felt that the reactions, both emotional and functional, to the long-term use of imperfect speech technology would be an important part of what was to be learned. This aspect of speech interfaces is often disregarded in studies.

Therefore, we enrolled four full-time student programmers from the speech group as our pilot users. They were experienced enough with window systems to have learned how to take advantage of windows and improved navigation. Although they had little, if any, exposure to speech recognition, they were certainly interested in its use; this made it likely they would try it enough to allow us to study their interactions.

Following an entry interview, our users were trained on how to use the system and given assistance in their selection of vocabulary names and initial configuration files. After this, they were observed for as much as two months.

We tracked Xspeak usage via extensive automatic logging, videotaping, and frequent short interviews. Logging recorded each word recognized and its recognition score, all Xspeak utility activities (such as retraining words and naming new windows), and all top level X window events.

Our users were all developing X Window System applications. Their basic screen layouts varied, but typically included a large editing window, a local terminal window for compiling the edited programs, a remote terminal window to receive mail and news, a local console window, Xspeak's window, and accessories such as a clock. The editing, terminal, and console windows were all text-based.

Empirical Analysis

We present empirical data on the tradeoffs between the mouse and voice below. Observations on the extended use of the system follow.

Timing

Input technologies are often compared on the basis of speed because of the belief that users will pick the most efficient interface. Therefore, we decided to join the fray, and we looked at the time required to complete a window transition using Xspeak versus the mouse. Table 1 shows the results. Speech was slightly slower than mousing for time to transition to another window. There was less variability in the spoken commands than in the mouse movements, perhaps due to differing window geometries and distances between windows.

		Times (secs)		
		Mouse	Xspeak	

User A	Mean	2.1	2.6	df=13,
				t=1.73

	Stddev	.6	.4	n.s. at .05

User B	Mean	2.0	2.5	df=14,
				t=1.66

	Stddev	.4	.9	n.s. at .05

Table 1: Timings

Table 1 shows the times, from videotape analysis, for each medium for two users. Times were measured from the start of the action (the user's hand moving off keys for the mouse and the start of speech for voice) to the first keystroke in the destination window (after the mousing motion). We excluded rejection errors (when a spoken name results in no action), transitions where the user clearly reads or thinks before typing, and all mouse transitions involving a button press.

Given the slowness of speech and the delays in recognition, we were not surprised that the mouse is faster. The difference is small enough that speech should be considered a viable input device.

However, these were optimal situations, and one might expect different behaviors in suboptimal situations. For Xspeak transitions, the user might experience rejection errors and have to repeat the window name. For mouse transitions, a user might need to move or lower several windows in order to find the desired window, or go through a sequence of mouse actions (such as handling a menu) to expose a buried window.

We were curious about how these more complex mouse motions would compare to speech times. In more realistic mouse interactions, times were as long or longer than speech. For user A, clicking on the title bar of a partially obscured window to raise it required a mean time of 2.8 seconds (s.d. = .3). Moreover, using a menu to expose a completely obscured window required a mean time of 4.2 seconds (s.d. = .6), a time substantially greater than speech. For user B, a double-click to raise a window required 2.5 seconds (s.d. = 1.0), a time that was comparable to speech.

Window Transitions

Window transitions, switching the pointer from one window to another, were the predominate navigational activity, and they occurred more often than we had anticipated.

	Session length (mins)	Xspeak chances	Xspeak use %	Mouse use hand on mouse %
User A	40	44	84	14
User B	40	73	63	22
User C	40	38	89	0
User D	20	22	100	–
Expert E	40	51	98	2
Expert F	45	49	96	2

Table 2: Xspeak Use within a Session

Users used Xspeak to navigate between windows about once per minute, based on our analysis of representative sessions on videotape. Within sessions, however, transitions were not evenly distributed; there were often flurries of window activity.

How often were window transitions made using Xspeak instead of the mouse when the window was named? As Table 2 shows, the percentages varied from 63% to 100%. (An Xspeak chance, in the table, is a window transition where the user could have used either Xspeak or the mouse. Window transitions to test Xspeak were not included.) When the user's hand was already on the mouse, the rate was substantially lower. These data were derived from videotapes of a single session with each user, and all window transitions in those sessions were included.

Recognition errors

Table 3 shows the *recognition* errors (when a spoken name results in no action) for six sessions analyzed in detail. Only recognizer errors are reported; user errors (e.g., speaking the wrong name) are not included.

Rejection errors varied considerably, from 16% to 58% of attempts. As mentioned, Xspeak was tuned to give rejection errors over insertion or substitution errors, and the consequences of an error were not very high. Table 3 presents the rejection data.

	Session length (mins)	Words spoken to recognizer	Rejection error rate (%)
User A	40	50	16
User B	40	70	47
User C	40	46	26
User D	20	28	43
Expert E	40	80	58
Expert F	45	62	23

Table 3: Rejection Error Rates

Because the rejection rates were higher than expected (an average of 35% with a standard deviation of 16%), we wondered whether the nature of the window navigation task was causing additional errors. So we conducted an additional study, involving three of the original six subjects, to obtain "pure" recognition rates. We asked the subjects to speak each window name, one after the other, for six cycles, at the beginning, middle and end of a half-hour of working session. In these results the navigation task did not have a statistically significant effect on recognition accuracy.

The high error rate may surprise most readers; there is a common perception that speech recognition (especially small vocabulary, speaker dependent, isolated word recognition) is a solved problem. Although experienced users with proper microphones and a quiet environment can achieve high accuracy, recognizers are not well behaved in natural settings. For example, Biermann [Biermann, 1985] reported error rates of 2% to 25% and Martin [Martin, 1989] reported rejection errors of 4% to 27%. With a head mounted noise canceling microphone (which both these studies used), these levels of errors are not unusual and simply reflect the variability of human speech without some practice. Our results are not inconsistent with these, given that we were using less robust microphones. (We must caution that comparing recognition results is risky without knowing the number and content of the words in the recognizer's vocabulary [Nusbaum, 1986].)

In our study, we chose not to use the head-mounted microphones traditionally used for speech recognition. Our informal evaluation of Xspeak with a head-mounted microphone gave recognition scores of over 90%. However, headsets are not, we feel, suited to everyday office use; they're uncomfortable over time, tend to slip, and interfere with common activities such as drinking coffee or answering the telephone. Although an even more directional microphone than the one we used might decrease background noise from fans and telephones, it would also be more sensitive to the speaker's exact position, and would also cause substantial rejection errors.

Users' reactions to their rejection rates varied considerably. As will be discussed below, users adopted a number of retraining and coping strategies.

Observational Analysis

Our users were a varied lot. Table 4 summarizes their working style and prior experience.

	Work Style	X Windows prior use
User A	typing with brief pauses	moderate
User B	typing with brief pauses	moderate
User C	typing with lengthy pauses	minimal
User D	typing with brief pauses	extensive
Expert E	constant typing	extensive
Expert F	constant typing	extensive

Table 4: Window System Use

Three of the four subjects programmed steadily; the other programmed for a while after a significant amount of thinking time. All of them had developed some navigation methods before using Xspeak; those with the longest exposure to windowing systems had developed the most extensive range of window behavior. For example, Subject D used the mouse extensively, e.g. for iconifying windows and for moving around inside his text editor.

In addition, we observed two of the coauthors, who are expert users. They both had substantial experience with windowing systems and used windows extensively for performing tasks in parallel.

Acceptance of Xspeak differed widely among our users, for a variety of reasons, as shown in Table 5. Subject A used Xspeak in the majority of his sessions for two months, and regularly asked other users to move if they were on an Xspeak workstations. He reported preferring Xspeak because it allowed him to have larger windows with more overlap. Indeed, his screen was typically much more cluttered at the end of the study than at the beginning.

Subject B liked fast machines, and Xspeak did not run on the fastest machines in the lab. After initial enthusiasm for Xspeak, he lost interest because of his poor recognition rates and began to use the faster machines exclusively. He also noted that if he already had his hand on the mouse, he preferred to continue using the mouse for window actions.

Subject C's navigation activity was minimal because of his low input rates. Moreover, he was fixed in his use of the mouse; he just did not find Xspeak to be sufficiently interesting to justify its use. Subject D used Xspeak for several weeks but thereafter his research work required hardware that conflicted with Xspeak, so we have limited data for him.

Expert E said using Xspeak allowed him larger windows with more overlap. When using Xspeak, Expert F allowed more window overlap, and typically used one to two additional windows than when limited to the mouse. Expert F, with his relatively high recognition, also believed that he had higher throughput with Xspeak, and therefore favored using it.

In summary, two users, including one author, preferred Xspeak to other input methods. Two users rejected Xspeak as being insufficiently interesting to outweigh their preferred system usage. One user found Xspeak interesting, but left the study for other equipment. The other user, an author, found Xspeak interesting, but his low recognition rates hampered his use.

Coping Behaviors

Poor recognition accuracy, in our opinion, was the greatest impediment to acceptance of Xspeak. The users who stuck with it had some of the higher overall recognition rates and developed successful strategies to overcome errors, as shown in Table 6.

Users retrained single window names (or the entire vocabulary), and calibrated audio levels in mixed amounts. Our most active user, A, had a strategy of retraining a single word when his recognition was low. He rarely trained the entire vocabulary (4 times, a

	Use of Xspeak	Recognition	Subjective Evaluation of Xspeak	Navigation Preference
User A	extensive	very good	enthusiastic	Xspeak preferred
User B	moderate	poor	not as useful as a faster workstation	fastest method
User C	minimal	moderate	not interesting	mouse motions
User D	minimal	poor	has potential	iconification
Expert E	moderate	very poor	positive although frustrating	all methods
Expert F	extensive	moderate	positive	Xspeak preferred

Table 5: User Experiences with Xspeak

to verify recognizer active:	calibrate the microphone
	speak each window name in turn
to verify recognizer discrimination:	speak some nonsense words
	speak some other window names
to improve overall recognition:	retrain the whole vocabulary
	verify that recognizer was responding
improve single name recognition:	retrain a window name
	rename a window

Table 6: Coping Strategies

minimum amount), but he often retrained a single word (109 single words in 79 sessions). On the other hand. two users with low recognition rates, B and E, showed the highest percentages of retraining the entire vocabulary and calibrating the recognizer.

All users, except one of the authors, had problems guessing which names would be most suitable for success with this particular recognizer. We expected our users to find suitable names for themselves with a minimum of training; this may have been unrealistic.

Although we began the experiment believing that navigation was a separate user task, our users did not distinguish between using the mouse for navigation among applications and using it for direct manipulation interactions within an application. As a consequence, Xspeak seemed incomplete to them.

Adaptive behaviors

We also observed users displaying behaviors that used speech in novel and creative ways. These behaviors would be present at any recognition rate.

We observed users speaking window names while in physical positions from which they could not operate a keyboard or mouse (such as answering the telephone). Even when sitting directly in front of the computer, users took advantage of the "hands-off" nature of speech input (see Table 7). Interestingly, both experts began to use Xspeak to "warp" the mouse. That is, they used Xspeak to move to the desired application and then used hand motions to fine-tune the mouse position.

physical freedom from keyboard	hands in lap, scratching head
	hands in motion to or from keyboard
	hand on mouse
physical freedom from workstation	yawning, stretching
	drinking coffee
	using the telephone
	looking in manuals
flexible window names	sounds with emotional content
	words unrelated to window function
	words representative of a window shape
	words from a foreign language

Table 7: Adaptations to Voice Input

Conclusions

The reader is cautioned against generalizing from our results. This set of case histories is very small. Furthermore, Xspeak was a prototype system under development during the evaluation, and simply did not work well during some periods.

Having said that, we believe we have observed several important behaviors on the part of our users, and some interesting characteristics. First, the individual differences we observed were substantial. For our most satisfied student user, speech input worked very well and gave him opportunities for creativity. This user has a strong preference now for speech input. For our least satisfied student user, speech held no attraction even though his recognition scores were quite good. For the other two students the results were mixed.

Our experiences suggest that users' preferences for an auxiliary speech interface may vary a great deal and, for some users, be unrelated to their success with speech recognition. Designers and evaluators of speech input systems should anticipate a wide range of user responses, depending upon the users' experience level, ability to achieve consistent recognition, their current strategies for managing windows, and the nature of their work.

Second, speech recognition is clearly still difficult to use, and expertise is required in setting up the recognition device, choosing a vocabulary, and training users. We decided against head-mounted microphones, and therefore we needed to have Xspeak's design minimize the impact of rejection errors. We observed several of our users developing interesting and successful coping strategies for times when the recognizer was not working well. Nonetheless, recognition rates are important and were the most consistently cited complaint about Xspeak. In a second version of Xspeak, we will address this by subsetting the user's vocabulary for recognition purposes.

Third, we observed that having Xspeak allowed users a greater range of physical motions and positions, since they were not so tied to the keyboard and mouse. Fourth, we saw some evidence for increased number and degree of overlap of windows while using voice. Longer term studies will be required to substantiate this point, however.

Finally, a speech interface to a window system needs to support direct manipulation interactions. As mentioned, our users did not distinguish between the use of the mouse within an application and among applications. We intend to add this capability in our second version as well, so that spoken words can invoke, for example, mouse button presses.

Acknowledgments

We would like to thank Gale Martin, for her insightful early discussion on evaluation methodologies; Wendy Mackay, for her invaluable help with our use of video as an evaluation tool and ongoing review of the work; Barbara Frederickson, whose comments on an early version of this paper were of great assistance; and Ralph Swick, for key advice on some obscure aspects of X.

References

[Biermann, 1985] A. W. Biermann, R. D. Rodman, D. C. Rubin, and F. F. Heidlage. Natural language with discrete speech as a mode for human-to-machine communication. *Communications of the ACM*, 28(6):628–636, 1985.

[Bly, 1986] S. A. Bly and J. K. Rosenberg. A comparison of tiled and overlapping windows. In *Human Factors in Computer Systems - CHI'86 Conference Proceedings*, pages 101–106, New York, 1986.

[Gaylin, 1986] K. B. Gaylin. How are windows used? Some notes on creating an empirically-based windowing benchmark task. In *Human Factors in Computer Systems - CHI'86 Conference Proceedings*, pages 96–101, New York, 1986.

[Gould, 1978] J.D. Gould. How experts dictate. *Journal of Experimental Psychology: Human Perception and Performance*, 4(4):648–661, 1978.

[Harper, 1985] R. Steve Harper. Voice data entry applications at Texas Instruments. In *Proceedings of the 1985 Conference*, pages 217–221. American Voice I/O Society, 1985.

[Martin, 1989] Gale L. Martin. The utility of speech input in user-computer interfaces. *International Journal of Man-Machine Studies*, 30:355–375, 1989.

[Nusbaum, 1986] Howard C. Nusbaum, Christopher N. Davis, David B. Pisoni, and Ella Davis. Testing the performance of isolated utterance speech recognition devices. In *Proceedings of the 1986 Conference*, pages 393–408. American Voice I/O Society, 1986.

[Nye, 1982] J. M. Nye. Human factors analysis of speech recognition systems. *Speech Technology*, 1:36–39, 1982.

[Schmandt, 1990] Chris Schmandt, Mark S. Ackerman, and Debby Hindus. Augmenting a window manager with speech input. *IEEE Computer*, August 1990.

[Visick, 1984] D. Visick, P. Johnson, and J. Long. The use of simple speech recognizers in industrial applications. In *Proceedings of INTERACT '84, First IFIP Conference on Human-Computer Interaction*, London, 1984.

[Wickens, 1981] C. D. Wickens, S. J. Mountford, and W. Schreiner. Multiple resources, task-hemispheric integrity, and individual differences in time-sharing. *Human Factors*, 23:211–230, 1981.