Developing Voice-only Applications in the Absence of Speech Recognition Technology

Anind K. Dey, Lara D. Catledge, Gregory D. Abowd and Colin Potts
Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta GA 30332-0280 USA
+1-404-894-7512
{anind, lara, abowd, potts}@cc.gatech.edu

ABSTRACT

In this paper, we describe an information access system with a voice-only interface. We outline a design process for generating guidelines for voice-only interaction in the absence of adequate speech recognition technology. Our usability studies make use of a "Wizard of Oz" scheme to replace the missing core technology.

Keywords

Voice-only interaction, speech recognition, Wizard of Oz

INTRODUCTION

Voice recognition and synthesis provide interesting and exciting interface alternatives for application developers. To date, voice has been used primarily to augment applications with an existing visual interface (e.g., VoiceNotes [6]). There are a couple of reasons why a voice-only interface to a system might be desirable. First, the application might require a hands-free mode of interaction. Second, telephone service is one of the few truly robust and ubiquitous network technologies, so it makes sense to extend information services away from the desktop by providing a telephone interface. We are developing an experimental system, which like MailCall [4], SpeechActs [8], and Wildfire [7], provides voice-only access to desktop and network-based information services.

We have investigated a wide range of information services so far enabling us to make generalizations and draw conclusions for develop some voice application guidelines. Our prototype system provides access to both network and desktop-based services, such as weather forecasts, stock market results, and personal messages (e-mail, voice mail and faxes). Access is initiated by a telephone call, after which the user issues voice commands for various services. The system responds with the information requested.

For instance, a user may call to find out if she has any messages. She finds that a colleague wants to meet that afternoon. Checking her calendar, she discovers that she is free and responds to the e-mail with a voice-attachment. Finally, she checks the West Coast weather for tomorrow because she has a trip to California planned. She sets a meeting reminder to herself and hangs up.

The scenario described above is idealized and is mimicked by many corporate concept videos. For the user, the interaction is natural. She switches between tasks seamlessly without losing the context of recent interactions. There are no excessive demands to remember mappings such as "press or say one for sports scores," familiar to most users of voice-based menu systems.

WORKING WITH SPEECH TECHNOLOGY

Two core technologies are required to move toward this ideal of voice-only interaction: speech synthesis and speech recognition. Speech synthesis is a readily available commerical technology. There are also commercial and academic systems to support speech recognition, but they are limited. The principal trade-off a designer of speech recognition systems faces is between versatility (speaker independence, size of vocabulary, continuous speech with real-time recognition) and recognition accuracy. A natural voice-only interaction needs to allow continuous speech over a seemingly unlimited vocabulary, from the user's perspective. Given this requirement, it appears that our demands on speech recognition far outstrip what is currently available.

Without the desired speech recognition technology it might seem impossible to empirically investigate other important aspects of voice-only interaction, such as user acceptance or satisfaction. We want to conduct realistic user studies that lead to guidelines for future developers of voice-only applications, but we do not want to wait for the technology to catch up. Herein lies the crux of the design problem: to design an application with sufficient functionality to determine user acceptance, without the use of this core technology. We decided, therefore, to use a Wizard of Oz approach, with a human operator performing the speech recognition.

METHODOLOGY

Three focus group meetings were held to discuss potential services and interaction techniques. Each focus group consisted of 6-8 participants and 2 moderators. The participants were given a brief description of the proposed application and a series of sample scenarios outlining how the application could be used. They took turns portraying

the user, while others took the parts of various information services. The interactions between the users and the services were recorded for future use. We used these meetings to develop a list of potential services and to get an initial glimpse of what an interaction might look like.

A large list of services was constructed using input from brainstorming sessions, a literature survey, and the focus groups. Of these, we initially prototyped four: stock market results, US weather forecasts, headline news, and messages (e-mail, voice mail, and faxes). The World Wide Web was the information source for all the services except for the message service.

The prototype was used in a series of usability studies. The purpose of these studies was to determine the usefulness of the provided services, build user grammars for future speech recognition integration, determine common navigation paths between and within services, and, of course, to determine the usability of the prototype. The usability studies were performed in two stages: an initial study with a small group of participants with an early version of the prototype and a second study with a larger group of participants using a mature version of the prototype. The results reported here refer to the second study.

Both stages of the usability studies were conducted in the same manner. The participants were given a short written description of the application and a list of tasks to perform. A participant telephoned the Wizard prototype and attempted to use natural language voice commands to complete the assigned tasks. The user was then given the opportunity to explore the system freely. The prototype generated a log file for each user containing time-stamped requests for data and time-stamped replies. A complete audio record was kept for later analysis. The final portion of the usability test was a short questionnaire.

OBSERVATIONS

Guidelines for developing voice interfaces are definitely lacking. There is some specialized research; such as in the area of question formatting [3] and using menu and list-style interfaces [5], but generalizable guidelines of the type familiar to GUI developers are largely absent.

General principles developed over decades of research obviously still apply; feedback is still important, as is direct control over an application's actions. If anything, working within a voice-environment highlights many important, well-known lessons. Though our work, we tried to apply these general principles in order to develop a first cut at guidelines that we hope will form the basis for further discussion and development.

Below are some initial observations that have come out of our usability tests.

Being Conscious of the Computer

The users were not told about the human operator, and were led to believe that a computer was controlling the entire system. This was done in order to obtain more realistic interaction examples. When participants knew that a human was "in the loop", they tended to phrase their queries differently: using informal language (e.g. "Can you please read me today's news?") and being very polite (saying "please" and "thank you"). When participants were unaware of the human operator, queries were phrased more as demands than requests (e.g. "Read headline news.") and more formal language was used, often mimicking the language used in the list of tasks provided.

A question to ask is whether knowledge of the computer implicitly leads a user to speak with a more limited vocabulary. We suspect this is the case, and if so, it actually makes the recognition problem simpler, and the key design principle is to create a human-computer conversation that implicitly limits user responses to a reduced vocabulary. It is a question of user perception; if users believe the system has a small working vocabulary, they will interact using a small vocabulary, to the speech recognition system's benefit. However speech recognition is made more difficult if users believe the system has a large working vocabulary because they will interact using a large vocabulary. This observation is illustrated below.

Adaptation and Learning

In general, users adapted their language during a single session with the system. When they experimented with more than just simple one-word commands and had success, they modified their speech to more free form queries. They learned quickly that the more verbose form of communication allowed them to navigate and access information more quickly.

We believe this was a result of having a human perform natural language speech recognition, rather than recognizing a constrained vocabulary. As users became more confident that the speech recognition "system" could understand a more conversational input language, they took advantage of it. This notion brings us back to the concept of user perception and the research question posed above: what is the relationship between user perception of how "intelligent" the computer is and the vocabulary they use? In order to understand this relationship better, further testing must be done.

Working Memory and Navigation

People have a limited working memory and this is even more important for voice-only interfaces than for GUIs. Without a graphical display to store information, users are forced to maintain necessary context in memory. When presented with long lists of unique options, users tended to take one of two paths. The first, and more common path, was to ask for the list to be repeated. The second, and unexpected, path was to write down the options, something we never intended our users to do.

The obvious solution is to avoid overloading the user with long lists. But this observation leads to the more general question of how to provide adequate support for users trying to navigate through the system. The support should enable users to feel in control of the interaction, and not cognitively overload them. The interface should be designed to work well with novice as well as expert users. We saw users that could adapt to the system and others that struggled. Obviously, one static interface will not be sufficient when dealing with a heterogeneous user population. Rather, a dynamic interface that changes with the user is needed. Prompts should be expanded when users require more help and should be succinct and non-intrusive when users are in control - part of a process known in the education community as scaffolding[2].

Importance of Feedback

Speed and the amount of time spent waiting was important to users. They did not want to wait a long time while data was being retrieved. A delay of more than approximately 5 seconds caused concern because they were given no interim feedback. Delays in voice applications cause more frustration than in GUI applications. The prototype provided no feedback telling users that it was retrieving data, whereas with a GUI application, the user may get feedback by looking at the display. Users require constant interaction in voice applications because they have no other method of control or feedback.

With an voice-only interface, only audio feedback can be used. Different cues can be used to tell a user when they should speak, when the system has or has not understood their requests, when it is fetching information (if long delays), and so on. Distinct and intuitive sounds are an effective way of providing audio feedback to users [1].

Granularity of Response

In the previous observations, we discussed how users feel a lack of control when waiting for system responses and the burden that voice output places on cognitive load. The same effect is felt when response granularity, the level of detail in system responses, is coarse. During our tests, we experimented with giving users varied degrees of response granularity. For example, with some users, we replied to a request for information about a particular stock with a 30 second long discourse on the current price, annual high and low, price-to-earnings ratio, and so on. With other users, we replied to the same request with a short reply simply stating the current price and the daily change, and providing options for obtaining the more detailed information, if desired. The second group of users were more satisfied than the first group. A goal is to provide the user as precise a response as possible, giving the feeling of greater control and revealing information progressively in order to reduce the demands on users' memory.

CONCLUSIONS AND FUTURE WORK

The goal of our project was to develop an information access system where voice was the only mode of interaction. The greatest design problem faced was the lack of a speech recognition system. Even without computer-controlled speech recognition, a useful system was developed using a Wizard of Oz approach. This enabled the testing of a variety of different interaction techniques and has pointed us towards future research avenues. As with GUIs, the important design considerations with voice-only interfaces deal with users' perception of the interface: how much control they have and how easy it is to use. The differences, of course, are in how these considerations are implemented. Our future plans include investigating the effects of user perception of system intelligence on vocabulary, scaffolding, feedback, and introducing constrained speech recognition along with more usability studies to measure the results.

ACKNOWLEDGMENTS

We thank Eric Buhrke and Jonathan Engelsma of Motorola for their support and collaboration. We would also like to thank Laura Burkhart and Robert Orr of Georgia Tech, for their continued support and efforts in this research.

REFERENCES

1. Albers, M.C., and Bergman, E. The Audible Web: Auditory Enhancements for Mosaic. In Proceedings of CHI '95 (Denver CO, May 1995), ACM Press, 318-319.
2. Guzdial, M. Software-realized Scaffolding to Facilitate Programming for Science Learning. Interactive Learning Environments, Vol. 4, No. 1, 1995, 1-44.
3. Hansen, B., Novick, D.G., and Sutton, S. Systematic Design of Spoken Prompts. In Proceedings of CHI '96 (Vancouver, Canada, April 1996), ACM Press, 157-164.
4. Marx, M., and Schmandt, C. MailCall: Message Presentation and Navigation in a Nonvisual Environment. In Proceedings of CHI '96 (Vancouver, Canada, April 1996), ACM Press, 165-172.
5. Resnick, P., and Virzi, R.A. Relief from the Audio Interface Blues: Expanding the Spectrum of Menu, List, and Form Styles. ACM Transactions on Computer-Human Interaction, Vol. 2, No. 2, June 1995, 145-176.
6. Stifelman, L.J., Arons, B., Schmandt, C., and Hulteen, E.A. VoiceNotes: A Speech Interface for a Hand-Held Notetaker. In Proceedings of INTERCHI '93 (Amsterdam, The Netherlands, April 1993), ACM Press, 179-186.
7. Wildfire Communications, Inc. Homepage. Available at http://www.wildfire.com.
8. Yankelovich, N., Levow, G., and Marx, M. Designing SpeechActs: Issues in Speech Interfaces. In Proceedings of CHI '95 (Denver, CO, May 1995), ACM Press, 369-376.