A Unified Design for Human-Machine Voice Interaction

Stefanie Shriver, Arthur Toth, Xiaojin Zhu, Alex Rudnicky, Roni Rosenfeld
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
+1 412 268 7678
{sshriver, atoth, zhuxj, air, roni}@cs.cmu.edu

ABSTRACT
We describe a unified design for voice interaction with simple machines; discuss the motivation for and main features of the approach, include a short sample interaction, and report the results of two preliminary experiments.

Keywords: Spoken dialog systems, speech, user interface

INTRODUCTION

For speech recognition to become widespread, users must learn how to speak to and interact with a variety of systems (information servers, handheld devices, transaction servers, household appliances, etc.). This includes knowing what vocabulary and syntax to use with each different system, as well as having some way of ascertaining a system's capabilities and limitations.

One solution to this problem is to use unconstrained, natural language dialog interfaces, in which a system is designed to respond to open, conversational input ("When's the first flight to New York Monday?" "Did my stocks go up?"). However, this approach can be problematic for both developers and users: a large amount of domain knowledge is required to sufficiently model possible user input, and the large vocabularies and complex grammars necessary for such systems can adversely affect speech recognition accuracy. Users may also experience problems if they overestimate a system's knowledge and ask it questions that it is not equipped to handle.

Another approach is to use machine-driven dialogs to guide users to their goals, but this is not much of an improvement over the touch-tone menu interfaces so ubiquitous in telephone-based systems. In these systems, the user is often forced to listen to a variety of options, most of which are presumably irrelevant to their goal. Interactions are slowed by this forced iteration of options at each step, and although frequent users may be able to speed up their interactions by memorizing the appropriate sequence of keypresses, these sequences are not valid across applications, and users therefore must learn a separate interface pattern for each new system used.

ANOTHER APPROACH

We have been working on an alternate paradigm for voice interface systems, called the Universal Speech Interface (USI). With this approach, users learn a set of strategies that help them explore and use any application that is designed using the USI protocol. The core features of the USI are a small set of keywords (less than ten) and a standard structure for input and output.

The USI keywords are designed to provide standard mechanisms for interaction universals, which we derived by analyzing several applications and application categories prior to developing the USI vocabulary. These universals include help, orientation, navigation, error correction, and general system interaction.

The standard structure provides principles governing the regularities in the interaction, such as *"the system will tersely paraphrase whatever part of the input it understood"* and *"input is always provided in phrases, each conveying a single information element."* Each application designer can choose how flexible the grammar should be for the individual phrases. This can range from a tightly proscribed format to unconstrained natural utterances. In the applications we have designed so far, in order to help improve recognition accuracy, the phrase grammar is quite stylized, as is seen in the examples below.

Our approach requires users to learn a specific interaction style. We found that the USI can be effectively taught in a five minute, one-on-one tutorial session, after which the user should be able to use any USI-compliant application. We are also experimenting with ways to effectively use more limited training time. One of our implementations is telephone-based, and includes a 90-second tutorial introduction (which advanced users can skip), which appears to do a reasonable job of telling users what they need to know in order to start using the system.

EXAMPLE

Our first USI application is a telephone-based interface to a database of current information about movies and theaters in the Pittsburgh area.

User (U) wants to know where Casablanca is playing:

1 *U:* Movie is Casablanca, theaters are what?
2 *Movieline (M):* Casablanca, two theaters: Showcase East, Waterworks Cinema.

User would like to find a comedy showing in Squirrel Hill:

3 *U:* Neighborhood is Squirrel Hill, **now what?**

4 M: Squirrel Hill. Now you can say: title is <dadada>, theater is <dadada>, genre is <dadada>, <ellsig>.

5 *U:* Genre is **now what?**

6 *M:* Genre can be comedy, drama, foreign, <ellsig>.

7 *U:* Genre is comedy, titles are what?

User inquires about the movie October Sky:

8 *U:* Theater is Cinematropolis, movie is October <timeout>.

9 *M:* <alert> Cinematropolis.

10 *U:* Movie is October Sky, theaters are what?

User wants to know what time Citizen Kane is showing at Waterworks Cinema:

11 *U:* Title is Citizen Kane, theater is, ummm, Water…. **scratch that!**

12 *M:* Scratched.

13 *U:* Movie is Citizen Kane, theater is Waterworks <timeout>.

14 *M:* Citizen Kane, Waterworks.

15 *U:* Times are what?

16 *M:* Six show times: 1:15, 2:45, 4:00 <ellsig>.

17 *U:* **More.**

18 *M:* 5:45, 7:50, 10:00.

The boldface words in the above dialog comprise the main USI keywords. They enable users to perform interaction universals such as correcting errors (**scratch that**), getting local help (**now what**) and navigating lists (**more**).

Input is provided in phrases, which reduces recognition and parsing complexity. It also provides discernable boundaries that help both the system and the user understand where errors occur. That is, for each phrase that was successfully parsed by the system, the USI returns a terse, value-only paraphrasing of that phrase (lines 13-14). If a phrase is unsuccessfully parsed, the user hears an alert, followed by any correctly recognized phrases (line 9). In the case where no phrases are correctly recognized, a more descriptive statement of the problem is issued by the system. If the system makes a recognition error that happens to parse correctly (e.g. "movie is Casablanca" is recognized as "movie is Citizen Kane"); the user will discover this error through the system's paraphrasing and can correct it using the **scratch that** keyword.

The use of phrases can also help mitigate errors: a user experiencing recognition problems can enter one phrase at a time and make sure it is successfully recognized before moving on to the next phrase. This of course slows down the interaction, but can be used as a fallback strategy when the system's recognition rate is low (for instance, if the user is speaking in a noisy environment or has a strong accent).

Another key feature of the USI approach is the use of non-lexical signals to "pack the audio channel." Using audio signals in interfaces has been shown to increase response times [1,2], and, if used in place of lexical descriptions, audio signals should often decrease the duration of output messages, which can affect overall task completion times. We believe that audio signals can also help universalize applications and reinforce learning across applications. In the example above, <ellsig> (lines 4, 6, 16) is an ellipsis signal (currently implemented as three short beeps), which indicates that the list has not reached its end. <alert> (line 9) is used to warn users that something in their input was not recognized; this is currently implemented as a beep. <dadada> (line 4) is a spoken placeholder signal, indicating that the user can fill in a value here.

PRELIMINARY RESULTS

In hope of finding a preliminary validation of our approach and obtaining useful feedback for refining it, we have performed two initial experiments.

In the first experiment, 15 subjects each used the USI movieline to find the answers to five movie information retrieval tasks. Our goal in this study was to make sure that the USI approach was usable in the most basic sense: did users understand the concepts of structured, phrasal input and keywords enough to complete basic tasks? All of the subjects were indeed able to complete tasks (although one needed the aid of a USI "cheat sheet"); nearly all the users formed correct USI-style commands within their first three utterances. One of the most informative results of this study was the need for explicit confirmation. This was not included in our first version, and resulted in users often not being sure how to correct errors since they were not sure what the system had and had not understood.

Our second experiment has so far provided us with anecdotal feedback only. Five subjects used the USI movieline and a natural language interface to the same database. Four preferred the USI interface, and most noted the system's transparency as its strongest asset.

FUTURE WORK

The greatest promise of the USI lies in cross-application transference of user skills. We therefore plan to build additional applications to test this effect. We also plan to perform more comprehensive user studies, focusing on design aspects such as how to present and navigate more complex data structures and how to optimize audio output.

ACKNOWLEDGMENTS: We thank the Pittsburgh Digital Greenhouse for providing seed funding for this project.

REFERENCES

1. Bussemakers, M. and de Haan, A. When it sounds Like a duck and it looks like a dog…, in *Proceedings of ICAD '00* (Atlanta, GA, April 2000), 184-189.

2. Leplâtre, G. and Brewster, S. Designing non-speech sounds to support navigation in mobile phone menus, in *Proceedings of ICAD '00* (Atlanta, GA, April 2000)