Contextual Push-to-Talk: A New Technique for Reducing Voice Dialog Duration

Garrett Weinberg

TR2009-062   November 2009

Abstract

We present a technique in which physical controls have both normal and voice-enabled activation styles. In the case of the latter, knowledge of which physical control was activated provides context to the speech recognition subsystem. This context would otherwise be established by one or more steps in a voice dialog initiated by a conventional, single ”push-to-talk” button.

MobileHCI 2009

Contextual Push-to-Talk: A New Technique for Reducing Voice Dialog Duration

Garrett Weinberg
Mitsubishi Electric Research Labs
201 Broadway, 8th floor
Cambridge, MA 02139 USA
+1 617 621 7547
weinberg@merl.com

ABSTRACT

We present a technique in which physical controls have both normal and voice-enabled activation styles. In the case of the latter, knowledge of which physical control was activated provides context to the speech recognition subsystem. This context would otherwise be established by one or more steps in a voice dialog initiated by a conventional, single "push-to-talk" button.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**], Input Devices and Strategies, Interaction Styles, Voice I/O

General Terms

Design, Human Factors

Keywords

Speech recognition, push-to-talk, voice dialogs

1. BACKGROUND: VOICE DIALOGS

1.1 When to listen

Interactive systems that afford a voice input modality rely on various strategies for distinguishing intended system input from background noise or background speech. So-called “always-listening” systems [1] employ a lexical analysis of any given buffered segment of audio, looking for keywords (e.g. “computer”) that are intended to “wake” the system into an active state. Other systems make use of input clues modeled after human-to-human discourse, such as gaze direction [6].

This work focuses on “sometimes-listening” systems – those that employ a “push-to-talk” (PTT) button that, when pressed, causes the system to process the subsequent segment of audio as intended speech input. In some implementations, the endpoints of the speech segment may be determined automatically by analyzing, for example, the amplitude or signal-to-noise ratios of the captured signal. In others, the user is required to keep the button held down until she is finished speaking, with the instants of button press and release serving as the segment endpoints.

1.2 What can be said

In PTT-based automatic speech recognition (ASR) systems, the “listen” action is invoked, voice input is given, recognition takes place, and the system interprets the recognized word or phrase to effect a transition to some application state. In addition to the recognized word or phrase, this interpretation may depend on the current application state.

The most advantageous capability that an ASR system offers in mobile or embedded environments—particularly in an eyes-busy, hands-busy environment such as the automobile—is the capability to easily choose a desired item from among many possible items without typing or excessive scrolling. However, due to the limited memory and CPU resources available in these environments, today’s embedded speech recognition engines impose some limitations on the user interface that do not necessarily apply to desktop or telephony server-based speech deployments.

For example, whereas a desktop or server-based system might be able to process a music-retrieval utterance such as “search artist Madonna” from any application state, a contemporary automotive deployment such as the Ford Sync [4] requires the user to switch into the music mode beforehand, by issuing the correct voice command (“USB” in this case) or by pressing the corresponding hardware button. If the user tries a music-retrieval utterance while in Phone mode, the command fails.

1.3 How long it takes to say

There is ample evidence that complex in-vehicle tasks such as music retrieval and destination entry negatively impact drivers’ tactical scanning and lane keeping behavior, especially as these in-vehicle tasks increase in duration [3, 5]. The Ford Sync product has been so well-received in part because its grammar and voice dialog design allows for such tasks to be kept short, with at most two steps required, for example, to retrieve a particular artist or album. Other recent commercial offerings require three or more steps.

2. CONTEXTUAL PUSH-TO-TALK

Our design proceeds from the realization that voice input need not be an afterthought when considering the physical human-machine interface (HMI) design. If the car or portable device in question is designed to have dedicated buttons for choosing screens or modes, can these buttons somehow be dual-purposed as voice input buttons? Instead of having a unique, single-purpose PTT button, couldn’t potentially *any* button or physical control be a “listen” control when activated in a certain way?

In such a design, the quick press of a mode button might switch to the mode in question, for example Navigation, Music or Contacts. A longer press or a double-press of the button could indicate the user’s wish to not only change to the mode, but also to

MobileHCI’09, September 15 - 18, 2009, Bonn, Germany.
ACM 978-1-60558-281-8.

immediately carry out a voice search in the mode; the paradigm in this case is "change to the mode, and find what I say."

Command—rather than mode—widgets can also be extended with a voice activation style. The omnipresent green "phone" button might, with an ordinary single-press actuation, bring up the Recent Calls screen. With a double-press actuation, it might cause the system to listen for voice input, in this case the phonebook entry that should immediately be dialed (e.g. "John Doe mobile").

Similarly, "play/pause" and "shuffle" buttons could accept voice modifiers. If the normal actuation acts as a simple toggle (play or pause, random playback on or off), the voice-enabled actuation would listen for the *target* of the operation (play *what*, shuffle *what*).

Whether it is applied to command or mode buttons (or other physical controls), the advantage of this design is that it eliminates at least one turn in a multi-turn voice dialog. In conventional approaches that use a single PTT button, the initial turn or turns are used to extract contextual information—for example information about the search domain of interest—that is then used to activate an ASR grammar appropriate to the next dialog turn. In this multi-PTT approach, the same contextual information is conveyed by the user's choice of button, allowing the system to skip directly to the later dialog turn.

The conventional and contextual-PTT approaches can be combined into a single system. Novice users may access any mode or function via a traditional, multi-turn dialog that leverages a dedicated PTT button. Advanced users would learn by tinkering or by reading documentation that other buttons *besides* the main PTT button also allow for voice input, when they are activated in a special manner. Having gained this knowledge, they are then empowered to bypass dialog turns and carry out their tasks more quickly.

3. PROTOTYPE AND EVALUATION

We have constructed an initial automotive prototype of this HMI to study its advantages and limitations. The prototype offers three domains—Navigation, Music, and Contacts—with unconstrained voice search in each domain. "Unconstrained" in this case means that users can say any words or phrases relevant to the desired items, in any order (the SpokenQuery voice search engine [7] provides this capability). GUI output and event processing are handled by a Java Swing application running on a PC, with an Optimus Mini Three OLED keypad [2] providing three themable input buttons. A single press of one of these buttons switches to the last active screen in the corresponding domain. Two consecutive presses of a button within a 300ms window activate voice search in the corresponding domain. During a voice search operation, a short tone sounds, a "Listening" screen appears that offers brief instructions about what can be said, and then after speech input, recognition and lookup, a result screen specific to the domain is presented.

We hypothesize that the affordance of domain-specific PTT buttons and the removal of the initial, domain-selection node in our voice dialog tree will both reduce overall task time and promote safer driving behavior (as measured by e.g. lane position variance). We intend to test these hypotheses in a driving simulator. To do so, we will compare the multiple-PTT approach discussed here with the use of a single, steering wheel-mounted PTT button that launches a traditional, multi-step dialog.

An important consideration in such a study will be to properly acclimate subjects to the physical position of the contextual PTT buttons (on the Optimus keypad) relative to the steering wheel. Hardware buttons were deliberately chosen (rather than e.g. a touchscreen) so that users' muscle memory can guide their fingers to the keypad and from button to button without the need to look away from the virtual roadway. The study protocol should therefore allow for sufficient training time such that muscle memory can begin to form.

4. REFERENCES

[1] Alewine, N., H. Ruback, and S. Deligne. Pervasive Speech Recognition. IEEE Pervasive Computing, 3(4): 78–81, 2004.

[2] Art Lebedev Studios. Optimus Mini Three keyboard 2.0. http://www.artlebedev.com/everything/optimus-mini/

[3] Barón, A. and P. Green. Safety and Usabilty of Speech Interfaces for In-Vehicle Tasks while Driving: A Brief Literature Review. Technical Report UMTRI 2006-5. University of Michigan Transportation Research Institute at Ann Arbor, Michigan, U.S.A.

[4] Ford Sync, 2007–2009. http://www.syncmyride.com

[5] Green, P. Driver Distraction, Telematics Design, and Workload Managers: Safety Issues and Solutions. SAE Paper 2004-21-0022.

[6] Myers, B., R. Malkin, M Bett, A. Waibel, and B. Bostwick. Fleximodal and Multimachine User Interfaces. In Proceedings of the IEEE 4th International Conference on Multimodal Interfaces. IEEE Press, 2002, pp. 343-348.

[7] Wolf, P., J. Woelfel, J. Van Gemert, B. Raj, and D. Wong. SpokenQuery: An Alternate Approach to Choosing Items with Speech. In Proceedings of the 8th International Conference on Spoken Language Processing. ISCA, 2004, pp. 221-224.