Object-Oriented Multimodality for Safer In-Vehicle Interfaces

Garrett Weinberg and Bret Harsham
Mitsubishi Electric Research Labs
201 Broadway
Cambridge, Massachusetts 02139, U.S.A.
+1.617.621.7547
{weinberg, harsham}@merl.com

ABSTRACT

Despite recent gains in the accuracy and flexibility of voice interfaces, speech-enabled in-vehicle information systems (IVIS) still impose a significantly higher cognitive load than vehicle operation alone [6, 7]. This results in degraded driving performance while carrying out common information-retrieval (IR) tasks such as finding a particular point of interest (POI) from a navigation database or a particular song from a music library. This paper proposes a reorientation of the IVIS interface around domain-scoped searches and contextual commands rather than around hierarchical menus and global commands. We believe that this design will reduce IR task time while also reducing cognitive load, thereby encouraging safer driving.

Categories and Subject Descriptors

H5.2. [**Information Interfaces and Presentation**]: User Interfaces – *Voice I/O; Input devices and strategies (e.g., mouse, touchscreen)*

General Terms

Performance, Design, Human Factors.

Keywords

Speech recognition, voice recognition, voice search, command-and-control, object-oriented interfaces, multimodality.

1. INTRODUCTION

1.1 In-Car Voice Interfaces

Numerous advancements have been made in the last few years in the flexibility and accuracy of automatic speech recognition (ASR) technology for embedded (often in-vehicle) use. Nuance and IBM have both introduced limited statistical language model (SLM) support into their embedded recognizers, enabling command-and-control utterances to vary significantly in their formulation (versus earlier, strictly finite state grammar (FSG)-based approaches) [10, 12]. Both Nuance and Novauris now offer one-shot voice destination entry (VDE) technology, wherein the house number, street, and city portions of an address may all be included within a single utterance [10, 8]. ASR error rates on difficult large-vocabulary recognition tasks (e.g., correctly identifying the spoken street name when searching among all the streets in Germany) continue to gradually improve for the ASR engines from all major vendors.

These fundamental technical advancements, however, have not been properly leveraged to improve the day-to-day usability of IVIS. This is due to the fact that most IVIS have interfaces which are built upon a hierarchy of system states. Functions are generally divided into groups of related actions which are only available from a particular node of the state tree. In order to perform an action, the user must navigate around the tree to the particular state where the action is available. This requires the user to maintain a mental model of the system state, and the available commands for each state.

This design leads to time-consuming stepwise interactions. Before being able to employ the one-shot address entry technology mentioned above, a user might first have to say “navigation” and then, “by address.” Take for example the currently popular voice-enabled Sync offering from Ford [5]. It can distinguish among the spoken names of thousands of song titles on a connected portable music player. However, unless the system is already in portable music player mode, users must first say “USB” before uttering a search phrase like “play track Nights in White Satin.”

Although car entertainment systems have grown from very simple radios with just a few modes (AM/FM) to complex computers with tens of modes and hundreds of functions, their human-machine interfaces (HMIs) still rely heavily on physical control elements such as buttons and knobs. Such elements often offer more efficient “command-and-control” than speech interfaces because they are familiar to drivers, are not prone to errors, and offer increased efficiency of use over time as motor memory develops. Most importantly, however, they shorten interaction times versus the step-by-step, hierarchical voice dialogs described above.

1.2 Multimodality

Some IVIS ease the hierarchy traversal process by allowing one to progress either by voice command or by manual controller (modality *equivalence* in the taxonomy given in [13]). That is, voice commands are available that equate to physical actions. For example, in the 2009 Acura TL, a user can advance from this vehicle’s “Search Music By” screen by either saying or manually choosing (via the multifunction input knob) the visible menu option for Album, Artist, Track, etc. In the Ford Sync, in addition to saying “USB” to switch to iPod mode as explained above, one can also press the USB button on the console (or, in certain vehicles, cycle through input sources by repeatedly pressing the Media button).

Though to our knowledge there have been no formal comparisons of systems offering modality equivalence to systems that do not, it stands to reason that the multimodal designs would improve driving and/or visual scanning behaviors. This is because such designs allow a user to proceed through a task using the modality that feels most appropriate—i.e. the least temporally and cognitively demanding—given the current traffic situation.

1.3 From Equivalence to Complementarity

Despite this presumed advantage, numerous studies have shown that even well-designed voice and multimodal IVIS interfaces do impose costs in terms of cognitive load, driving performance, and visual scanning behavior ([6, 7], survey in [1]).

We contend that this is in part due to their limited application of multimodal interaction principles. In the taxonomy cited above, Vilimek et al. (borrowing from Martin [9]) discuss how information from individual input modalities can be fused to increase the throughput or decrease the ambiguity of interactions [13]. Whereas with the modality *equivalence* detailed above, "several modalities can be used to accomplish the same task," with modality *complementarity*, "the complete information of a communicative act is distributed across several modalities." The combination of information from multiple modalities provides higher throughput and thus decreases task time.

1.4 Experience from Implementation of Modality Complementarity

We recently implemented a prototype IVIS which incorporates modality complementarity [14].

In this prototype, it is possible to search for an item from the user's music collection (song, artist, or album), a Point of Interest (POI) from the nearby area, or a person from the user's phonebook. The choice of which of these "domains" to search is established by the user's choice of a particular push-to-talk (PTT) button among several possible PTT buttons. Each PTT button is uniquely associated with a domain of interest. These buttons each activate a listening tone, letting the user know she may speak her search terms. The top results of the search are presented visually, and a manual controller can be used to select the desired search result.

Rather than the spoken search terms' context being conveyed by initial steps in a dialog—or by a carrier phrase, as in the "play track Nights in White Satin" example above—the search terms' context is conveyed by the tactile modality, i.e. by which of the several PTT buttons the user has pressed. The input operation is incomplete without the contribution of both tactile and voice modalities; each complements the other.

This design could be thought of as an inversion of Bolt's classic "Put That There" interface [2], in which the referents for the spoken deictics "that" and "there" are resolved via pointing gestures. In our case, the *put* becomes a *get*. You tell the system from where you want the spoken item to be retrieved by pointing to (and pressing) a tangible representation of the kind of item it is (a button labeled with a textual or graphical representation of that item type).

In a usability evaluation conducted in a driving simulator, first-time users required 40% less time to carry out IR tasks using the multi-PTT approach than they did using the conventional, single, state-aware PTT button. They also performed more consistently in the task of following a lead vehicle, and reported a preference for the multi-PTT approach for daily use in their cars. This study is reported in [15]. One of the interesting results of this study was that although the version of the interface offering modality complementarity (multi-PTT) was less distracting to users in terms of cognitive load and driving performance than the single-PTT variant, it was still measurably worse than unencumbered driving.

2. OBJECT-ORIENTED MULTIMODALITY (OOM)

We suggest that although modality complementarity is helpful in reducing cognitive load, a system design based on a state tree may be a fundamental limitation. In order to use such a system, a user must first map the desired action to a system state, then recall how to transition the system into the desired state, all prior to beginning a dialog with the system. In addition, before beginning to speak, the user must mentally model the system's current state and decide how to express the command or function in the system's currently active vocabulary.

We contend that a radical redesign of the interaction model may be more intuitive for users. In our new model, which we call Object-Oriented Multimodality (OOM), the user thinks of and specifies the object first (e.g., "Thai restaurant" or "Maureen Peterson"), and then, in a separate utterance, says what she wants to do to that object (e.g., "go there" or "call her cell phone").

In this model, we treat all user actions as IR tasks, where the IR task is divided into two distinct phases. In the first phase, the user searches for and retrieves an object to act on. This search leads to a second, object-oriented phase, in which the object that has been retrieved can now be used. The actions available in the second phase depend on the kind of object that has been found (a POI, an album, a contact from the address book, etc.).

This results in a find-then-activate interface that inverts the thought process involved in a conventional command-and-control interface, wherein the user must first formulate a command phrase describing what they want to do (taking into account that some commands might not be available in the system's current state), and then provide the target of the formulated command, all in one utterance.

The following provides more details on each phase of an OOM interaction.

2.1 Search

Current IVIS systems typically have a set of buttons that allow the user to choose between the main areas of functionality (NAV, PHONE, MEDIA, etc). These buttons can be overloaded for use as "contextual" PTT buttons of the sort suggested in our prototype implementation (by adding, for example, press-and-hold or double-press actuation styles). The content domain is established by which button the user chooses as a PTT in order to begin the interaction.

In many content domains (for example POI, music, and contacts), it makes sense to present the best matches to a spoken query in the form of a relevance list, especially if the scores assigned by the decoder and/or the IR engine all fall within the same narrow range. Unless an audio-only interface is used to present this match list[1], there will necessarily be a GUI metaphor that conveys which item within the result list is currently active, selected, or in focus. This can be accomplished using anything from a simple highlight box placed around a textual description of the selected item to a revolving "carousel" of high-resolution item icons or images, as demonstrated, for example, by Audi and nVidia at CES 2010 [3].

While the selection box or focal lens could theoretically be moved using voice commands like "next" or "next page," most users find that approach clunky. Industrial designers have spent years honing physical controls such as rocker switches and rotary dials to make them pleasurable and effective to use for exactly this task. Why reinvent the (mouse-)wheel?

Instead, our design encourages brief navigation within the result list using these time-tested manual controls. If the user finds herself scrolling through more than five or ten items, a voice-based search repetition or refinement may be warranted. However, with a relatively unambiguous query like "Jimmy's Steakhouse" and proper filtering or re-scoring of results based on such factors as proximity (in the case of POI) and history/frequency of access (in the case of music and contacts), the desired item is likely to appear at or near the top of the match list the majority of the time.

2.2 Action

Once the desired item has been activated via the tactile modality, the user issues a voice command with the focal item as this command's implicit referent. This is similar in spirit to Oviatt's map-based multimodal mock-up where users could, for example, circle a house while saying a command like "show photo" [11]. The operand of the action (the house, in this case) is established via manual input, while the action itself is established by spoken input.

In our proposed system, the commands that are available in the action phase would depend on the kind of item that was in focus. For a POI item, these commands might include "call" and "show on map." An album might support the actions "play" and "shuffle," for example. One might be able to dispatch such commands as "text" or "call at home" to a contact item.

SLM-based free-form command technology such as that described above could be leveraged to allow for more "natural" contextual commands such as "please play track three from this album" or "I want to call her on her cell phone." The performance of such technology would be greatly enhanced by the absence of globally-scoped commands; we can make the engine's job easier by activating a small SLM that is limited to the domain of discourse (music, POI, etc.) and designed to assume the presence a focal item in that domain.

2.3 Further Details

While OOM as described covers the lion's share of IVIS functions, automotive voice user interface (VUI) experts will be quick to point out that neither digit-based dialing nor address entry fit neatly into the hypothetical contacts and POI search domains we have mentioned. The former might best be addressed by employing the multi-function button paradigm introduced in [14] and [15]. The green "call" button found on the steering wheels of many Bluetooth-enabled cars generally performs a "redial" or "recent calls" function with a single tap. When the user instead double-taps this button, the system could issue a listening tone and accept a string of spoken digits that will be dialed.

Address entry takes a bit more cleverness. Keeping in mind that if POI and phonebook-based destination entry are implemented well enough, users will seldom need to enter an address by house number, we propose a compromise whereby street/city pairs are included in the POI index and retrieved in the same way as businesses. House number or intersection info can then be provided to the retrieved pseudo-POI (a given street/city combination) as explained above. Think of the way one tells a taxi driver one's destination, starting with the more granular information and then providing the house number or intersection later, perhaps only upon nearing the destination: "Peachtree Street in Atlanta. Number 180." Users should find such a design sufficiently intuitive.

Other voice commands in a contemporary IVIS enable the manipulation of various system settings or preferences, which, despite their infrequency of use, inflate the size of ASR grammars and hence decrease the accuracy of recognition. We propose incorporating these functions *en masse* into their own IR domain by indexing the human-readable description of each function from the system's user manual. Each indexed document corresponds to a given setting or application state, and retrieving such a document is equivalent to executing that command or jumping to that application state. This "settings" IR domain would receive a dedicated button of its own, just as the other, content-oriented domains described above.

It should also be mentioned that while this design discourages globally available voice commands, some vital contextual commands such as "help" and "back" should be available no matter what kind of item is in focus.

3. DISCUSSION AND FUTURE WORK

In the near future we plan to validate our approach by conducting iterative prototyping and usability evaluation. While [15] offered an initial indication that users indeed prefer to implicitly specify search domains via their choice of button rather than by stating the name of the domain first, in that study there were only three active domains. This did not include the "settings" domain proposed above, and there was no action phase required to complete an IR task. We need to integrate these aspects into our prototype and see if user satisfaction remains high.

In addition, there would seem to be a break-even point where the profusion of IR domains, each with its corresponding PTT button, results in a multimodal UI that is just as cluttered and opaque as today's VUIs with global commands for everything under the sun. But whether that break-even point is four domains or fourteen domains remains to be determined.

While switching to an OOM design alone is unlikely to result in the ideal automotive HMI that is no more distracting or cognitively demanding than unencumbered driving, the only way to verify that it is indeed a step in the right direction is to empirically evaluate driving behaviors and eye glance durations

[1] Auditory-only presentation of long lists/menus places high demands on working memory [16].

within a simulator or a suitably instrumented vehicle. We plan to conduct this work in the coming months.

4. REFERENCES

1. Barón, A. and Green, P. 2006. Safety and Usability of Speech Interfaces for In-Vehicle Tasks while Driving: A Brief Literature Review. Technical Report UMTRI 2006-5. Ann Arbor, MI: University of Michigan Transportation Research Institute.
2. Bolt, R. A. 1980. "Put-that-there": Voice and gesture at the graphics interface. SIGGRAPH Comput. Graph. 14, 3 (Jul. 1980), 262-270. DOI= http://doi.acm.org/10.1145/965105.807503
3. Cunningham, Wayne. 2010. Audi A8 handwriting recognition hands on. CNet Australia. http://www.cnet.com.au/audi-a8-handwriting-recognition-hands-on_p7-339300398.htm#vp. Retrieved 10 May, 2010.
4. ETSI EG 202 191. 2003. Human Factors (HF); Multimodal Interaction, Communication and Navigation Guidelines. ETSI, Sophia-Antipolis Cedex, France. http://docbox.etsi.org/EC_Files/EC_Files/eg_202191v010101p.pdf. Retrieved May 6, 2010.
5. Ford Motor Company. 2010. http://www.syncmyride.com. Retrieved 26 May, 2010.
6. Garay-Vega, L., Pradhan, A.K., Weinberg, G., Schmidt-Nielsen, B., Harsham, B., Shen, Y., Divekar, G., Romoser, M., Knodler, M., Fisher, D.L. 2010. Evaluation of Different Speech and Touch Interfaces to In-Vehicle Music Retrieval Systems. *Accident Analysis & Prevention*, 42, 3 (May 2010), 913-920.
7. Graf, S., Spiessl, W., Schmidt, A., Winter, A., and Rigoll, G. 2008. In-car interaction using search-based user interfaces. In *Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1685-1688. DOI= http://doi.acm.org/10.1145/1357054.1357317
8. Hunt, M. and Kim, Y. 2006. Phonetic Techniques for Achieving High Accuracy in Spoken Access to Very Large Lists. In *Proceedings of 2006 AVIOS Speech Technology Symposium*. New York: Applied Voice Input Output Society.
9. Martin, J.-C. 1998. Types of Cooperation and Referenceable Objects: Implications on Annotation Schemas for Multimodal Language Resources. In LREC 2000 pre-conference workshop, Athens, Greece.
10. Nuance Communications, Inc. 2008. Nuance Introduces Natural Language Speech Suite for Navigation & Automotive Vendors. http://www.nuance.co.uk/news/20080305_suite.asp. Retrieved 25 May, 2010.
11. Oviatt, S., DeAngeli, A., and Kuhn, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, United States, March 22 - 27, 1997). S. Pemberton, Ed. CHI '97. ACM, New York, NY, 415-422. DOI= http://doi.acm.org/10.1145/258549.258821.
12. Sicconi, R., White, K. D., Ruback, H., Viswanathan, M., Eckhart, J., Badt, D., Morita, M., Satomura, M., Nagashima, N., Kondo, K. 2009. Honda Next Generation Speech User Interface. SAE World Congress & Exhibition, April 2009.
13. Vilimek, R., Hempel, T., and Otto, B. 2007. Multimodal interfaces for in-vehicle applications. In *Proceedings of the 12th International Conference on Human-Computer Interaction*: Intelligent Multimodal Interaction Environments (Beijing, China, July 22 - 27, 2007). J. A. Jacko, Ed. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 216-224.
14. Weinberg, G. 2009. Contextual push-to-talk: a new technique for reducing voice dialog duration. In *Proceedings of the 11th International Conference on Human-Computer interaction with Mobile Devices and Services* (Bonn, Germany, September 15 - 18, 2009). MobileHCI '09. ACM, New York, NY, 1-2. DOI= http://doi.acm.org/10.1145/1613858.1613960
15. Weinberg, G. and Harsham, B. 2010. Contextual push-to-talk: shortening voice dialogs to improve driving performance. In *Proceedings of the 12th International Conference on Human-Computer interaction with Mobile Devices and Services* (Lisbon, Portugal, September 7 - 10, 2010). MobileHCI '10. ACM, New York, NY.
16. Wickens, C. D., Sandry, D. and Vidulich, M. 1983. Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors*, 25, 227-248.