Nuance Voice Control for Automotive

Enabling Safe and Usable In-Vehicle Messaging

Garrett Weinberg and Lars König, Nuance Communications

Tomas Macek and Jan Kleindienst, IBM Czech Republic

May 2011

Contents

1. Introduction

1.1 You Talk (and Drive), It Types

Dictation was the initial “killer app” for automatic speech recognition (ASR) technology when it came to mass-market desktop computer platforms in the mid-1990s. This capability allowed for the creation of documents without touch-typing knowledge or even necessarily the manual dexterity required to type.

Systems of this early era struggled with continuous streams of words spoken at normal conversational pace, and they required extensive training to adapt their acoustic models to the voice of a particular speaker. Today’s systems have neither of these limitations.

Given these advancements and the increasing ubiquity of mobile devices with a persistent Internet connection, there is an increased demand on the part of car makers to offer dictation solutions for the vehicular context. These solutions generally involve the car head-unit capturing the input audio (and optionally running an ASR front-end) and subsequently sending the compressed audio or speech features over the paired mobile phone’s data channel to a server-class machine that actually performs the recognition and returns the result.

1.2 Connectivity, At a Price

The underlying goal of such deployments is to safely enable the messaging and social networking services that users have become accustomed to without requiring eyes-intensive touchscreens or keypads for text input. If voice-enabled solutions are unavailable for such use cases, rising distraction-related crash rates suggest that drivers will use their mobile phones directly to input and send text messages, despite this being explicitly outlawed in many countries.

For example, according to [9] about 30% of drivers surveyed in Australia sometimes entered a text message using their mobile phone while driving, and one out of six drivers did so regularly. This is despite the fact that drivers realize that receiving and especially sending messages is one of the most distracting in-vehicle tasks [26].

1.3 Driver Distraction Concerns

Given this situation, system designers must keep driver distraction foremost in mind when developing automotive UIs. Secondary aims are to minimize task completion time and maximize input quality. Significant effort has already been devoted worldwide to the analysis and reduction of in-car devices' impact upon drivers' attention. [5] offers a systematic review of the literature on this topic. A study by the AAA Foundation [18] compares several types of distraction. In recent years a set of standard assessment tools and metrics has been developed. The most widely used among these are the Lane Change Task (LCT) [13] and the car following task [12]. Some of these metrics are even entering the international standardization phase (e.g., ISO proposal #26022 for the LCT).

Meanwhile, progress is also being made on understanding how the human mind works and what the relevant limitations of human cognitive system are [1], [3], [4], [25].

2. The Text Entry Task and Driver Distraction

There are several studies illustrating the distraction potential of manual alphanumeric data entry tasks (e.g., [14]). In this section we will cover the two types of distraction that are relevant in this context, visual and cognitive. We will also briefly discuss how a system's inherent qualities can impose cognitive load upon the user.

2.1 Visual Distraction

Of primary concern are those occasions when drivers take their eyes off the road to attend to the device or navigation system display with which they are interacting. Unencumbered drivers naturally tend to keep their away-from-road glances shorter than about two seconds [17], but drivers who are carrying out visually-demanding secondary tasks have been shown to dwell on displays for far longer than that [21], [22], [2], putting themselves and other drivers at serious risk, especially at higher vehicle speeds (a car traveling at 120kph covers about 66 meters in two seconds).

Figure 1 summarizes the results of a recent study [11] on in-car dictation. Two distraction measures, MDev (Mean Deviation) and SDLP (Standard Deviation of Lateral Position) were collected for 12 users as they performed the LCT (Lane Change Task) at a simulated 60km/h and also conducted a secondary task. The secondary task was in all cases the composition of several text messages. The messages were composed using a standard cell phone, using a dictation system equipped with screen, and using a dictation system without a screen. The results clearly show the impact of a secondary task on drivers' performance. The dictation systems allowed drivers to perform significantly better than they could when composing messages on a cell phone. The best performance among the secondary-task sessions was observed for the dictation system without a display (“eyes-free” in the graph).

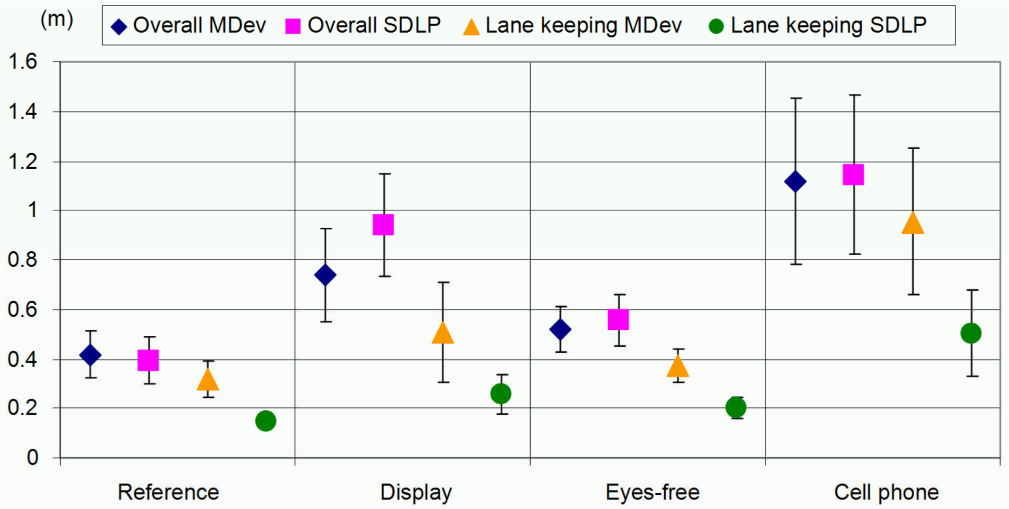


Figure 1: Average MDev and SDLP in [11], with 95% confidence intervals.

2.2 Cognitive Distraction

Aside from visual stimuli, the user’s attention is influenced by other factors. The terms cognitive capture and perceptual tunneling [20] refer to the phenomena by which one’s focus tends to narrow, reducing awareness of peripheral motion cues, when one becomes cognitively engaged (for example, by a conversation in which one is participating). In the case of message dictation this problem may be particularly pronounced because one tends to think about the content and context of the message one is composing—the conversation, as it were—while also paying attention to the navigation system’s visual interface for carrying out that composition (if indeed there is any such visual interface).

Wickens’ widely cited multiple resource management theory [25] holds that humans can only process a finite number of perceptual inputs at any given time, especially if more than one of those inputs occupies the same sensory channel (vision, hearing, touch, etc.). As the number of concurrent inputs in a single channel increases, one becomes cognitively loaded and one’s task performance suffers. This relates both to performance in the primary task, in this case operating a vehicle, and in any secondary tasks, i.e., operating an in-car device or application.

As driving is a highly visual task, Wickens’ theory—as well as the practical experience of academic and industry experts—dictates that we must be particularly careful how we construct the graphical user interface (GUI) for dictation as well as all other applications supported by the head-unit. Considerations for the design of such GUIs will be discussed in greater detail below.

2.3 Inherent Cognitive Load

Besides an overload of stimuli in the same sensory channel, the other primary sources of cognitive load are the factors inherent in the system itself, that is, in its design and realization. The system should communicate its state and expectations clearly so that the user does not have to expend mental effort deducing what he should do in a given situation. With regard to the dictation domain in particular, as suggested above, the user must keep in mind the already-composed content of the message as well as how to formulate the next phrase or sentence. The additional cognitive load imposed by the interface he is using should be kept as minimal as possible.

It is important to keep in mind, however, that one person’s confusing mess is another’s comfort zone. Tradeoffs must be made that keep systems accessible to the novice or occasional user but not annoying or condescending to the expert or frequent user. A common solution to this problem is to offer an “expert mode” setting that alters the behavior of the user interface without fundamentally modifying the system’s core behavior.

A novice mode/expert mode separation is workable only up to a point, however. Pervasive divergence between two interface styles across an entire deployed system can lead not only to development and quality assurance headaches, but more seriously, to an inconsistent, poorly branded user experience.

3. User-Centered Design and In-Car Dictation

User-Centered Design (UCD), at its core, means making the target user’s *needs*, *desires,* and *capabilities* paramount at every stage of the system design process [23]. We will briefly address each of these tenets with an eye toward the in-car dictation task.

3.1 Needs and Desires

It’s arguable whether anyone *needs* to send an SMS or update his Facebook status while on the road, but users certainly *want* to—especially users in the 18- to 25-year-old demographic. Other user groups might be motivated to, for example, respond right away to an important e-mail from a customer or manager. Each of these groups wants to interact with a system that is consistent and intelligible in its design, and that meets their expectations with regard to the capabilities it supports and the degree to which it resembles systems they have used in the past.

Since dictation in the car is a new use-case, users' expectations may not be fully formed. However, users might reasonably expect that, besides the text entry capability itself, a dictation solution will provide a set of voice commands to correct recognition errors and to choose recipients for the message, as well as actually sending it. The requisite command grammars must be designed with both users' expectations and the entire system's self-consistency in mind. If commands have a similar grammatical structure from domain to domain and function to function, it will be easier for users to predict what the acceptable command set might be for a new domain such as dictation, without needing to consult a user manual or tutorial.

3.2 Needs and Capabilities

The other key tenet of the UCD paradigm is consideration of the user's *capabilities* or *limitations* with regard to system interaction and task performance. In the context of website design this often refers to accessibility: making the site available to and enjoyable by persons with a physical disability such as limited vision. In the automotive context, on the other hand, the primary capability considerations have to do with safety and suitability. Taking into consideration the user's limited psychomotor and cognitive resources while performing the primary task (operating the vehicle), the system must nevertheless enable the user to accomplish her secondary task (dictating a message, etc.).

A balance has to be struck between satisfying a user's wants and addressing his limitations in the vehicular context. For example, a user may *want* to view the entire text of the message she is in the process of reading or composing. Providing this feature, however, might be inappropriate to her capabilities and limitations as a driver. She may *want* a graphically complex, text-heavy display, but she *needs* to get home safely.

3.3 Considering the System's Capabilities

It is crucial that the system's underlying capabilities only bubble up to the level of the user interface when these capabilities enhance, rather than diminish, the overall user experience. In other words, the user will not be impressed that the system accomplishes some great technical feat if it does so in an unpleasant or annoying manner. For example, if the system can accurately recognize any street name 100% of the time from among tens of thousands of possibilities, but takes ten or fifteen seconds to return its result, this is less acceptable to the average user than a system which can correctly recognize the spoken street name only 95% of the time, but returns its result in, say, three seconds.

The system's interface should be designed with sensitivity to the underlying technical limitations without making these limitations too visible. For example, it would jeopardize recognition accuracy to play a prompt such as "say any command" in a situation where only commands from certain domains are present in the grammar, or to offer confirmation feedback using a non-preferred variant of a command with high acoustic similarity to another command (leading the user to perhaps try this more confusable variant on her next attempt).

4. Dictation Design Challenges

Dictation itself is one of the easiest speech-recognition tasks for users to comprehend: "You talk, it types," as the marketing tag-line goes. Anyone can imagine him or herself leaning back in an office chair—or a bucket seat, as the case may be—and rattling off sentence after sentence of unbroken prose, with the computer following diligently along like a secretary in a 1950s steno pool.

In reality, things are not so simple. Real speech is full of discontinuities (hesitations, pauses for breath) and disfluencies (mispronunciations, "ums," "ers," etc.) that humans unconsciously ignore but that computers cannot necessarily ignore. Thus it is vital that dictation systems offer one or more mechanisms to correct misrecognized words.

In a desktop environment this is an easier problem to solve. The user has a large screen on which to view the recognized text in real-time, and can make corrections using voice commands, the keyboard, the mouse, or some combination of these three.

However in the vehicular environment, when the user's eyes are busy and attention is split, it can be difficult to detect when the system has made an error, and there is no mouse available to click on a misrecognized word and no keyboard with which to retype it.

4.1 The Detection Problem

As mentioned above, it may be problematic to display the entire recognized text once a user has stopped speaking and the system has had a chance to process it. The lateral scanning required to read and parse a long phrase or sentence will likely keep the user's eyes away from the road for greater than two seconds, especially considering the substantial amount time necessary to refocus one's eyes on the screen when they have been focused on the road ahead, and vice versa.

So should we simply remove the visual display of the recognized text entirely? That may not be the right answer either. If we present the recognized text only audibly, this places high demand on the auditory channel, which is by definition a serial one and therefore attentionally demanding; one cannot thin-slice one's attention to a stream of audible speech as one can to a visual stimulus (by glancing to the screen and back to the road repeatedly, in rapid succession). If you stop paying attention to the spoken text for a moment to attend to another stimulus, you risk losing comprehension of the content. In addition there is the problem of detecting homophone substitution errors (or near-homophones: words that are insufficiently distinct from one another as pronounced by the text-to-speech (TTS) system). Clearly it's not desirable for the TTS engine to spell aloud every word that has a homophone. It would be annoying to hear "It was great—spelled G-R-E-A-T—to meet—spelled M-E-E-T—you—spelled Y-O-U—today," since no one but a practical joker would ever intend to dictate "it was grate to meat ewe today." However there will occasionally be homophones or near-homophones that make sense in the context of the sentence (as could be deduced from their language model weight), and these could benefit from this form of disambiguation by spelling aloud. An example might be "I'll have to check the wait/weight."

In general, voice-only dictation is prone to producing outgoing messages that have more errors. It is also harder to correct the errors that are spotted. On the other hand, a design such as this encourages the user to keep his eyes on the road. Augmentation of the audible interface with a well-designed screen (requiring only quick glances) seems like an attractive option, and will be presented in greater detail below.

4.2 The Correction Problem

While there may be some situations in which a few errors in the outgoing message are tolerable, there are other situations where correcting any dictation errors is simply mandatory from a user's point of view. Think, for example, of an informal message to your friend regarding plans for the evening versus a message to your boss about the status of an important project.

Besides possible homophone substitutions as mentioned above, the system may miss uttered words, insert undesired words, make capitalization/acronym errors, or create undesired compound words (or separate words where a compound word is desired). Another class of error to be corrected has nothing to do with the performance of the recognizer. Sometimes the user simply misspeaks or changes her mind about what she wants to say.

In any of these situations, before a phrase, word, or letter/capitalization can be corrected, it must first be selected. There are both voice and tactile means of accomplishing this, and even some ongoing research into using the driver's gaze to select the word that should be corrected [10]. We will recommend techniques for item selection and correction below, after providing some design guidelines for voice interfaces in general and for the dictation use-case specifically.

5. Design Recommendations

5.1 Introduction

We first present some cornerstones of voice user interface (VUI) design that are informed by the research discussed above, and which apply in the general case as well as specifically in the case of in-car dictation.

Any detailed design of a VUI depends heavily on the constraints of the operating environment. This includes both the available computing resources (memory size and CPU throughput) as well as the input and output hardware incorporated into the system (for example the number, placement and size of buttons, knobs and displays). In other words, a voice, graphical, and tactile/haptic UI that works well in one car may be less appropriate or less functional in another.

Taking this into consideration, in the closing chapter of the paper we will mention some instances where our recommended design depends on the presence of a particular hardware capability, and discuss what the options are when this assumption is not met.

While we hope this document establishes a set of "best practices," UI design is a highly subjective matter, and approaches always need to be proven through user testing and refined through iteration. If particular aspects are modified in isolation—for example the wording of a particular voice command or the flow of a particular sub-dialog—a wider view on the whole system's VUI/GUI must be maintained, and conceptual consistency maintained.

5.2 General VUI "Cornerstones"

5.2.1 Help

The most fundamental tenet of VUI design is the inclusion of a useful help feature. With strictly-graphical applications a help function or mode could be seen as making up for a shortcoming in presentation or visual design; in other words the layout, labeling and relative positioning of the graphical widgets on the screen should by themselves convey how the system is supposed to be used. On the other hand, with an entirely voice-based application, or an application intended to be used with minimal reliance on the screen (as for example when driving), the user has little to no idea how to achieve his goals. Particularly the first few times he uses the system, he will require explicit assistance on how to carry out tasks using the VUI. Without a help feature of some kind, he will not know either what is possible to say or when (i.e., in which application states) the available commands can be said. VUIs should therefore sensitively but firmly guide users toward the supported utterances and patterns of interaction.

5.2.2 Listening and Processing Status

One stumbling block for naïve users when they first encounter a VUI is understanding when they can speak. Many are unaware that systems typically require an explicit button press to initiate a voice dialog—or sometimes before every individual utterance in a dialog. Particularly if a push-to-talk (PTT) button press is *not* required for each separate utterance, the system must make it clear that it has opened the microphone and that it is the user's turn to speak. This is best accomplished by playing a short, pleasant (but insistent) mid- to high-frequency tone just before the microphone is opened.

This tone should be accompanied by a clear, unmistakable visual indicator. We recommend something that can easily be perceived in the user's peripheral vision as she focuses on the road; for example a change in screen background color or a pulsing highlighted border around some or all graphical elements. It is also beneficial to provide a visual indicator that the amplitude of the captured speech signal is within the range required by the ASR engine.

If the delay in the recognition of longer phrases or sentences is significant (more than two or three seconds), it is important to let the user know that recognition is still in process. For these instances, we recommend supplementing the “listening” status message/visualization with a “processing” one.

5.2.3 Control Commands

Certain essential control commands such as Back (or Correction), Cancel (or Start Over), and Help should be available at every step of every dialog. This improves the consistency of the interface and helps users orient themselves when lost. Furthermore it is crucial that these commands, if spoken, be executed immediately without any processing delay or confirmation steps.

5.2.4 Prompting

With regard to the way the user will be prompted to speak, there are several aspects to consider. First of all, how often will the aforementioned listening tone be played? Some systems play the tone only once, at the beginning of each dialog, and some after each individual prompt to indicate the handover of control to the user. We recommend the latter approach as it makes it clearer to the user when the microphone is open and speech is being recognized. However, the use of a separate sound indicating transfer of control (as distinct from the start-of-dialog sound) could also be considered. This is further explained in section 5.3 below.

5.2.5 Multimodality

Broadly speaking, systems wherein the user can interact using more than one of her senses are referred to as multimodal systems. The ringing mobile phone that also lights up and vibrates is a familiar example of multimodal output. Multimodal input can involve gaze or eye tracking and pen or gestural input, but most often in the automotive context we refer to a multimodal system as one that processes simultaneous or sequential voice and manual (tactile) input while displaying information on a central LCD screen.

The question for UI designers is to what degree the text or graphics displayed on a given screen mirror the available voice commands while that screen is active. The “say what you see” approach dictates that any visible text also be “speakable” text. This offers the advantages of VUI and GUI parallelism and consistency, easing the user’s comprehension of what she can say at any given time. On the other hand, this design can encourage a lot of heads-down reading of the screen, with the dangers inherent in that. Furthermore, users may not be motivated to memorize the available commands if they have a live “cheat sheet” available at all times.

At the other end of the spectrum is a design in which the GUI and VUI are kept completely independent of one another. In such systems, the particular screen or module being displayed at any given time does not constrain or scope the available voice commands; you can for instance issue a navigation command while viewing your music library. This approach allows maximum flexibility and efficiency of interaction, but this flexibility is a double-edged sword. For example, a user may think he can begin to dictate text at any time rather than first entering the correct mode via the “send message to John Doe” command. Also, it may not always be possible to keep all content search domains active at all times, especially if the system supports, for example, point of interest search (among hundreds of thousands of POI) alongside song search (among tens or hundreds of thousands of songs on local or Internet sources). Without a VUI that matches the content type hierarchy depicted in the GUI, users’ expectations may not be met.

In terms of a multimodal system’s non-voice modality, it is important to consider a design that is appropriate to the automotive environment. Touchscreens have become widespread in cars after gaining popularity in industrial and consumer electronics (including mobile phones). However, automotive GUI designers must take great care to ensure that touchscreen widgets (buttons, scrollbars, etc.) are large enough so that they can be targeted quickly and accurately, with an absolute minimum of visual attention. Traditional knobs and buttons do not require as much attention in the targeting phase—especially once the driver has accustomed herself to the car’s cockpit and can therefore feel her way quickly and easily to the tactile control in question without looking down at all.

5.2.6 Settings

A perennial challenge in user interface design is the accommodation of different classes of user. Novice users, those encountering a given system—or indeed any voice-activated system—for the first time, have very different needs and demands than expert users.

Feedback from deployed systems shows that if users fail within the first few tries to successfully accomplish their intended task using a voice interface, they will cease to use this modality at all in future attempts, despite any theoretical gain it may offer in terms of safety or efficiency. Therefore it is worth incorporating a “beginner mode” to ease such users into comfort and familiarity. Beginner mode should have the following characteristics:

- It should mention that beginner mode can be turned off within system settings.
- Upon first use of the PTT button, it should explain the importance of speaking clearly and waiting for the listening tone.

- It should offer more verbose prompts with explicit instructions and/or examples of what can be said in a given command domain.
- It should incorporate a longer “no speech detected” timeout, allowing for more hesitation as users formulate their commands.
- It should be split into domains and should automatically turn itself off when a certain number of successful dialogs have taken place within a given domain. For instance, beginner mode for voice destination entry (VDE) should turn off when the user twice succeeds in entering a destination.

Upon deactivation of “beginner mode,” the system defaults back to normal mode, where prompts are succinct and “no speech” timeouts are shorter. It may also be worth adding an “expert mode,” where prompts are omitted entirely or played at a faster rate.

5.3 The Dictation VUI

Let us now focus more specifically on design recommendations for the dictation case. In [11] the authors compare two prototype GUIs for the dictation task. A design where the entirety of the dictated text is displayed on the screen caused significantly more degradation in driving performance than a design where no text at all was displayed. However, the number of spelling and content errors in the outgoing message was significantly higher when there was no visual feedback (leaving only the audible readout of the recognized text). Note that in neither case were steering and lane maintenance as compromised as they were when users composed the same text using their mobile phones.

To the extent that there is a desire to correct outgoing messages (see section 4.2 above), the challenge is to provide an interface that allows users to recognize and fix errors without spending too much time looking away from the road.

We glean from [11] that during the dictation itself, no textual output (indeed, few graphics at all) should be shown on the screen. The user should be concentrating on driving, primarily, and on the content of her message, secondarily. Feedback should come in the form of text-to-speech read-out that follows each snippet of recognized text. The following chart offers an example of our recommended dialog flow for text input. Earcons (short, unique sounds conveying a particular meaning—the audible equivalent of an icon) [6] are used to audibly convey the transfer of the conversational “floor” from the system to the user.

	User	**System**
1.	<Push-to-Talk button>	<earcon: ready>
2.	Send a text message to Michael Johnson	*Voice A:* Texting Michael Johnson. Please start dictating. <earcon: your turn>
3.	Hi Michael comma I will be about fifteen minutes late for the meeting period	*Voice B:* Hi Michael <short pause> I will be about fifteen minutes late for the meeting <earcon: your turn>
4.	<Silence>	*Voice A:* Please continue dictating or say, for example, 'send message,' 'correction,' or 'help.' <earcon: your turn>
5.	Feel free to start without me period	*Voice B:* Feel free to start without me <earcon: your turn>
6.	Send message	*Voice A:* Would you like to hear the message before sending it? <earcon: your turn>
7.	Yes	*Voice B:* Hi Michael <short pause> I will about fifteen minutes late for the meeting <longer pause>. Feel free to start without me <longer pause>. *Voice A:* Please say 'correction,' 'send,' or 'cancel.' <earcon: your turn>
8.	Send	Message sent. <earcon: end of dialog>

Table 1: Sample dictation dialog

As this example illustrates, if recognition is performing well—or at least well enough for the audience of the intended message—the entire interaction can take place with the user's eyes never leaving the road and her hands never leaving the steering wheel.

An additional aspect of the voice output modality is illustrated here. In order to make it easier for the user to distinguish the composed text being read back from the "status" prompts that solicit further interactions, two different voices are used. The user should be able to configure the system such that the voice used to represent his or her composed text is a male or female voice, as per preference.

5.4 Dialog Pacing

The following chart makes explicit an assumption about the flow of the dialog in steps 3 through 5 above:

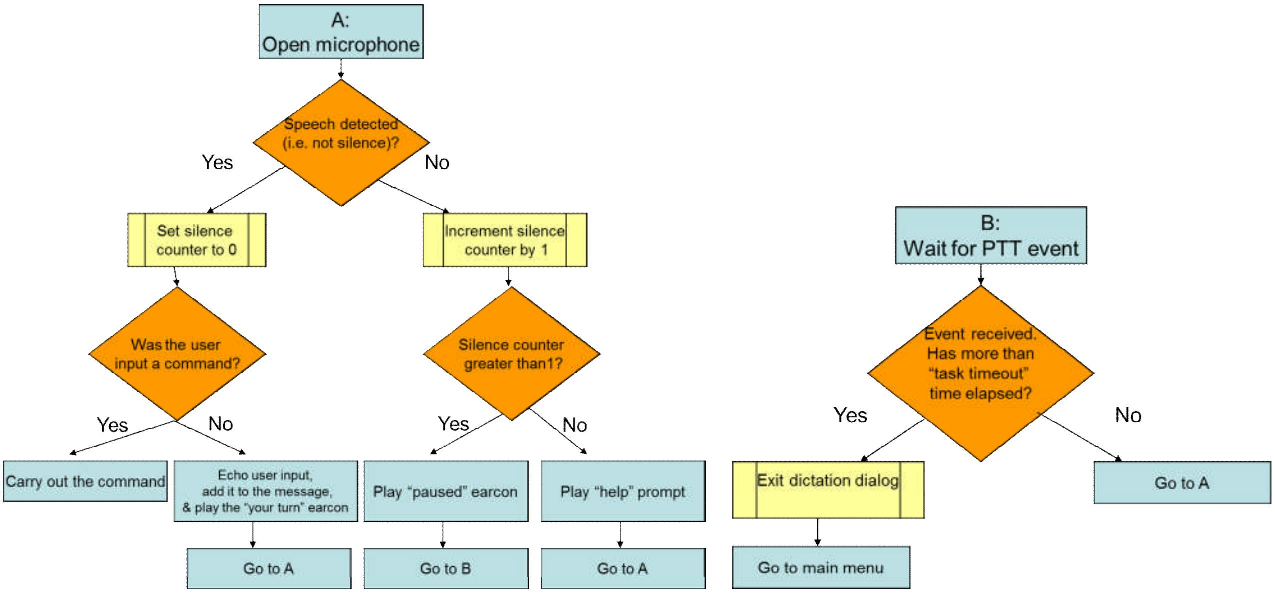

A:
Open microphone
Speech detected (i.e. not silence)?
Yes
No
Set silence counter to 0
Increment silence counter by 1
Was the user input a command?
Silence counter greater than1?
Yes
No
Yes
No
Carry out the command
Echo user input, add it to the message, & play the "your turn" earcon
Play "paused" earcon
Play "help" prompt
Go to A
Go to B
Go to A
B:
Wait for PTT event
Event received. Has more than "task timeout" time elapsed?
Yes
No
Exit dictation dialog
Go to A
Go to main menu

Figure 2: Pausing a dialog. "Task timeout" is a large value (at least two minutes), corresponding to the time it can be expected that a user will remember her place in an ongoing voice dialog if she has had to interrupt it to attend to driving.

Voice-enabled in-car systems must be "polite" in the sense that they seamlessly take to the background when the driver has more important business to attend to, namely maneuvering the car in a demanding traffic situation. To this end, the dictation dialog and all others that comprise the VUI should automatically pause themselves with no loss of progress when silence is detected twice in a row. An earcon as well as a visual symbol (for example a change of icon or color scheme) should make it clear that the dialog is in a paused state and can be resumed at will, ideally by the user's pressing the PTT button again. Only after sufficient time has elapsed in the paused state that a user is likely to have forgotten what she was doing should the system automatically cancel the active dialog.

Obviously, this “task timeout” behavior has to be crafted with care, such that it acts as a convenient accommodation to limited short-term memory rather than a forced restriction upon the user. If the timeout is made too short, the user will become frustrated and potentially distracted when he tries to resume a dialog and finds the system has defaulted back to its resting state.

5.5 The Desirability of Correction Mode

As stated above, a certain number of recognition errors during dictation are inevitable. However, given the audibly- and cognitively-intensive nature of correcting them without a GUI, and the visually-intensive nature of correcting them with a GUI—and the potential psychomotor demands of doing it either way—one could reasonably ask whether in-car dictation systems ought to offer correction capabilities at all. Omitting them—at least while the vehicle is in motion—might send a clear message that car and system manufacturers care about safety above all. “If precise textual accuracy is necessary, it’s best to send your message when the car is stopped,” might read the explanation in the driver’s handbook.

On the other hand, carmakers’ marketing departments might find this a tough pill to swallow. A BlackBerry-addicted middle manager might opt for a competitor’s car if that car’s navigation system has the most full-featured messaging component.

For this reason we have decided to design as safe a correction interface as possible, and leave the choice up to the OEMs and their electronics suppliers as to whether to enable the correction interface while the car is in motion. Driving simulator and eventually test-track based usability studies might provide data that informs this decision.

5.6 Implementing Correction Mode

5.6.1 N-best Lists?

In thinking about a correction capability for dictation results, UI designers familiar with statistically-based speech recognition techniques usually arrive at a common approach. If the recognizer has alternate hypotheses about what was said, why not offer the user the ability to correct errors by choosing among these hypotheses? This seems at first blush like a reasonable idea, but there are several potential drawbacks:

- The recognition engine may offer N-best lists only at the phrase level rather than the word level.
- Either phrase- or word-level N-best lists may not contain the item actually spoken, due to out-of-vocabulary errors.
- Phrase-level compounding errors (insertion of separate words where a compound word was desired, or vice versa) are difficult to correct using word-level N-best lists.
- Similarly, the recognizer may have inserted an undesired word or omitted a spoken word. With an N-best list-centric correction approach, it is not immediately clear to the user how to correct these errors.

An even more fundamental problem with the use of N-best lists in the automotive context is their visually-intensive nature. While this might be mitigated by presenting the lists audibly rather than visually, one must then contend with the homophone/near-homophone problem mentioned above (section 4.1). Moreover, the longer an audible N-best list is, the more difficult it is to retain its items in one’s short-term memory—especially if these items are acoustically similar, which tends to be the case with N-best lists. A reasonable compromise might be to give the user the option of requesting only the most probable alternative rather than delivering the entire N-best list.

For the reasons listed here, a correction design reliant upon N-best lists should be introduced only after thorough and careful consideration of their benefits and drawbacks.

5.6.2 A Multimodal Approach

Rather than an N-Best list-centric, highly visual design, we recommend a multimodal approach to the correction problem that blends the strengths of the voice and visual modalities while overcoming (or at least masking) the weaknesses of each.

As discussed above, visually representing the entire dictated text is problematic. Adults are conditioned to read passages of text in a continuous left-to-right (or in some cultures, right-to-left) fashion. Because of the difficulty in re-orienting oneself within a stream of text after having glanced away, we are tempted to read as much text as is presented on the screen at once—even when the same information is available through another sensory channel, as it was in [11]. This may lead to unacceptably long glances away from the forward roadway.

For this reason, we recommend not showing any text at all while the user is actively dictating new message content (as shown in Table 1 above). Let us call this mode *dictation mode*. In cases when the recognizer is performing acceptably well and/or the user does not care about minor recognition errors, the entire task can be carried out using this auditory-only interface; the driver's eyes need never leave the road.

However there will still be many cases where users will wish to correct any minor errors in their outgoing messages. To accomplish this they initiate in some fashion a *correction mode* (for example by saying "correction" at step 4 of Table 1's dialog). To allow for efficient correction without overloading the visual channel, we propose a "snippet view:" a window—preferably located in the instrument cluster area of the dashboard for better "glanceability"—that includes only a small chunk of the dictated message rather than the message in its entirety.

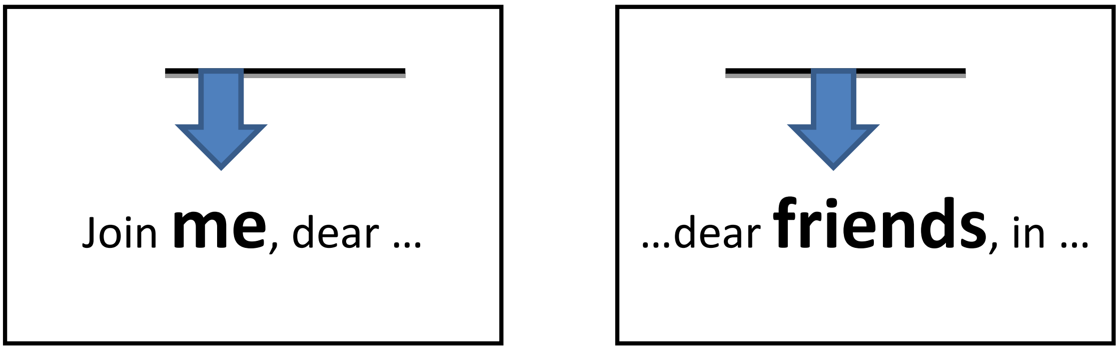


Figure 3: Correction mode. At left, the second word of the dictated text has been selected using, for example, a multifunction knob. At right, the fourth word has been selected. When a word is selected, the text-to-speech engine reads it aloud.

At the center of this view, the word with the current focus appears in a larger and/or bolder typeface. Upon entering correction mode, the first word in the most recently dictated chunk of text will be in focus. This focal point can be moved by, for example, turning the center console's multifunction knob or by pressing the left/right directional buttons located on the steering wheel. When a new word is brought in focus, it is read aloud by the TTS engine. This style of readout—as opposed to, say, a guided dialog that advances through the text automatically, word by word—allows the user to pace the correction task as the demands of driving allow.

To aid at-a-glance comprehension of the focal word's context within the sentence, snippet view shows one to two words preceding and one to two words following the focal word. These words are rendered using a smaller typeface. Naturally, at the beginning of the sentence there will be no preceding words shown, and at the end of the sentence no following words.

Snippet view could be thought of as a sliding window onto the dictated text. The focal word always remains at the center of the view, and a visual indicator above or below the snippet (such as the black horizontal line and vertical blue arrow in Figure 3) indicates the relative position of the snippet within the entire text.

5.6.3 Scoped Voice Commands

With the focal word highlighted in an enlarged or bolded typeface, users will easily comprehend that voice commands they use will be scoped on that word; "delete" clearly applies to the focal word rather than to the entire message. Similarly, the command "replace with XX" has a natural referent (the focal word), lessening the chances of repeated misrecognition errors (a "replace *this* with *that*" command is unhelpful if *this* is repeatedly misrecognized). Other useful word-scoped commands might be "insert XX" (which would add the provided word or phrase immediately after the focal word) and "capitalize" or "all caps."

If, in correction mode, the user simply dictates text without prefacing it with a command, this text is interpreted as the XX in "replace with XX." To instead append text to the end of the message, it is necessary first to return to dictation mode by saying, for example, "resume dictation" or "back."

5.6.4 Correcting Spelling

Even with a straightforward means of replacing or inserting words, there will still be words it is impossible for the recognizer to understand. This is the classic out-of-vocabulary (OOV) problem that has plagued voice recognition systems since the outset: the recognizer will only return words that appear in its dictionary or lexicon. While the OOV problem can be mitigated by using a large lexicon and keeping it as up to date as possible as new words enter the language, it will never be possible to capture all the possible words a user might say. For example, what if a user wants to dictate a message about a new song on the Top 40 charts whose singer has a completely fabricated name, perhaps one that is spelled using a combination of letters, numbers and symbols (e.g., "2Pac" or "Ke$ha")?

Because of the OOV problem it is essential that a dictation UI offer the ability to correct individual letters as well as words. Taking as a the starting point the snippet view shown in Figure 3, we can imagine something analogous for letters.

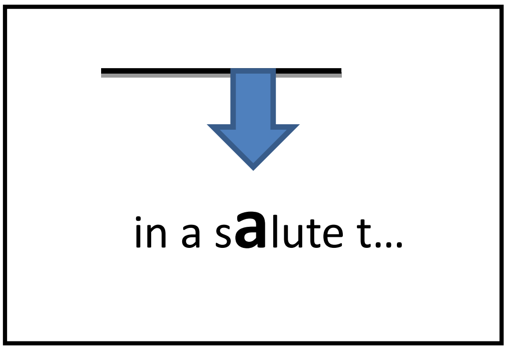


Figure 4: Spelling mode. The user interface is analogous to that shown in Figure 3, but the scope of the issued tactile and voice commands is the currently focused letter rather than the currently focused word.

The user could enter *spelling mode* by issuing a voice command such as “correct spelling” from either dictation mode or correction mode. Alternatively, she could “zoom in” as it were, from correction mode to spelling mode by tapping the “select” button on the steering wheel or activating the multifunction knob’s central momentary switch.

Spelling mode has an entirely parallel design to word-correction mode. Spinning the multifunction knob or using left/right directional buttons moves the focus back and forth among the letters—or more precisely stated, it moves the stream of text back and forth underneath an unmoving, central focal “lens.”

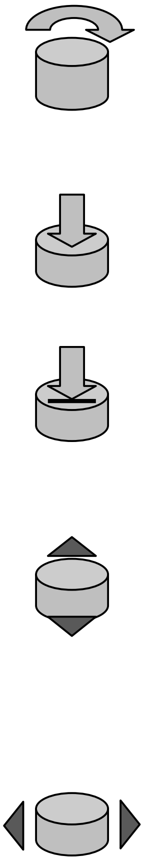
“Delete,” “replace with,” and “insert,” commands apply here as well, though because of the acoustic similarity of spoken letters, a phonetic spelling alphabet should be supported as an additional input option. Let us assume that a hybrid local/remote recognition approach is used for commands and dictated content, respectively. In this case, a spelling alphabet standardized to the particular language of deployment (for example, military and aviation’s Alpha-Bravo-Charlie for North American English) could be incorporated into in the same embedded grammar that recognizes local commands. At the same time, arbitrary “non-standard” phonetic spelling utterances such as “replace with A as in Apple” or “insert A as in Adam” could be recognized using the off-board dictation server. System feedback in the case of low recognition confidence could guide users toward the canonical variant supported by the embedded recognizer (“Did you say C, Charlie? Or D, Delta?”).

5.6.5 Tactile Input Considerations

For word correction mode, care must be taken when including a tactile fallback solution reliant upon lists or drop-downs, for the reasons stated in section 5.6.1 above. However, a tactile solution for letter entry seems to be a warranted fallback because spelling by voice can become unacceptably time-consuming if there are more than a handful of recognition errors.

While it is outside the scope of this article to recommend a particular tactile technique for letter input, there are good automotive predictive-text implementations that employ directional buttons, multifunction knobs and/or center-console touchpads. We do not recommend touchscreen-based solutions for letter entry due to their high visual demand.

Taking the popular centrally-mounted multifunction knob as an example tactile input/haptic output device, the following is one of many possible command mappings that support the dictation VUI design discussed above:



Turn: move to the next or previous word or letter, in word-correction or spelling mode, respectively. If in dictation mode, launch word-correction mode.

Short push: toggle between dictation, correction, and spelling modes.

Long push: delete the currently selected word or letter.

Tilt up/down: replace the active word or letter by its next or previous alternate, based on the recognition result (for words) or the predictive text model (for letters). No drop-down box is shown; the replacement is made in-line.

Tilt left/right: same as **Turn**.

Hold left/right: skip to the beginning or end of the message.

Figure 5: Example tactile interaction scheme for correction mode.

This example mapping presumes the presence of a dedicated PTT button near the multifunction knob. If this PTT button is not present, it may make sense to assign, for example, a double-press of the central momentary switch to the PTT action.

6. Additional Details and Conclusion

6.1 Deployment Considerations

Many aspects of our recommendation must be tailored to the particular deployment scenario. Just as no user is alike and each brings her own particular capabilities and limitations, no car's interior is exactly alike, and each interior enables certain capabilities and imposes certain limitations upon the system. Each car will have different types and numbers of screens, buttons, knobs, and other human-machine interface (HMI) elements, and these differences must be respected in the design and implementation of multimodal UIs for dictation.

6.1.1 No Screen, No Buttons

First let us consider how we would adapt the presented design to scenario where the screen (if any) and buttons are not available to the dictation application. This is a challenging environment, but one in which telematics service providers might be eager to play (with the entire service being offered via telephony using a paired Bluetooth device).

Dictation mode, once activated from the system's root menu, would be unchanged, as it is already an eyes-free mode. Correction mode and spelling modes, however, would require a significant overhaul, as there is no longer any tactile affordance for moving back and forth within the just-recognized text. With no screen, identifying errors becomes a major challenge as well. As discussed in "The Detection Problem" above, the challenge in designing any auditory-only correction mode would be determining when the system should spell homophones or near-homophones aloud as it reads back the recognized text. If you set the spell-aloud threshold too low, you risk too many errors creeping into the outgoing message. If you set it too high, reviewing the message becomes too tedious and time-consuming.

Because of these issues, in the buttonless, screenless deployment scenario we recommend a focus on short messages and the omission of navigation within the text entirely. The normal correction pattern would be "delete and dictate," which might be augmented by selected additional commands (read the whole message, replace X with Y, etc.).

6.1.2 No Screen

The "eyes-free" deployment scenario is one in which tactile controls other than the conventional "listen button" (a.k.a. PTT button) are available, but there is no display to provide visual feedback to the driver. This could be either because the vehicle lacks such a display or because it has been switched off in order to minimize visual distraction.

This scenario resembles the one described above, except that the additional tactile controls (for example those defined in section 5.6.5) enable the driver to perform navigation as well as editing of dictated text fragments. As in the other screenless case, special care must be taken to help the user detect errors (for example, by automatically identifying and warning about possible homophones). Note that in [11], the variant matching this deployment scenario caused less driver distraction than the variant equipped with a screen ("if there is a screen and its content changes, users will often look at it regardless of whether the displayed information is necessary for them").

6.1.3 Small Screen, Fewer Buttons

A more common deployment scenario is one in which the screen and buttons are indeed available to the speech/dictation application, but limited in some fashion by cost or distraction concerns. Let us say, for example, that the primary screen is a two-line, text-only LED rather than the widescreen LCD commonly found in higher-end vehicles.

Luckily the proposed design accommodates this style of screen well, as only a small amount of text is displayed at any given time, and only during correction and spelling modes. The orientation diagram (line plus arrow) shown at the top of the "snippet view" in Figure 3 is entirely optional, and could be represented in a more compact fashion, for example as a brighter or thicker dot in a series of dots.

In addition, let us imagine that there is only one button available to the dictation application, rather than a directional pad or a multifunction knob (fewer buttons and knobs, after all, makes for a cleaner, less overwhelming HMI). Most of the proposed design could remain as-is; we would simply need to be more creative with the way we employ this single button, for example by using long presses or double presses and combination with voice commands. One possibility is the following:

Action	**Result (dictation mode)**	**Result (correction and spelling modes)**
Short press	Interrupt any current readout of recognized text and open the microphone for additional dictation or a command (including "correction" and	Advance to the next word or letter. If at the end of the message text, cycle back to the beginning.

	"spelling" commands).	
Long press	Same as short press	Open the microphone for a command (including correction commands such as "delete" and "replace with" as well as mode-switching commands such as "resume dictation").

Table 2: Possible command mapping for a one-button dictation VUI.

6.1.4 Multiple Screens, Additional Inputs

Higher-end systems offer additional possibilities for the HMI. Principal among their affordances are multiple high-resolution displays, perhaps including head-up displays (HUDs). We are currently studying a dual-display dictation GUI design that supplements the instrument cluster display's "snippet view" with a whole-message view in the larger "center stack" display. The question is whether drivers will still be tempted to look at the full-text display if a snippet view is available and is closer to their optimal line of sight for driving. If experiments still show significant visual distraction in this configuration, the full-text view on the center-stack display could be made available only while the vehicle is parked.

In terms of optimizing glance times and durations, a very interesting possibility opened up by luxury-tier vehicles would be the presentation of the correction GUI in a HUD rather than in the instrument cluster. Such displays have shown promise in the past for reducing visual distraction during text-heavy multimodal tasks [24].

Higher-end vehicles are also some of the first to feature touchpads in their center consoles, which—when combined with accurate handwriting recognition technology such as T9 Write [15]—are logical extensions to the spelling mode presented here. Touchpads of course have applications outside of message composition as well, for example making common voice search tasks such as POI and music retrieval more flexible and multimodal.

6.2 Conclusion

We have motivated and presented a multimodal user interface that enables safe and usable in-vehicle messaging. Using a design such as this, carmakers might add SMS, e-mail, instant messaging, and social networking functionality to navigation and entertainment systems in order to make their vehicles as a whole more attractive to today's hyper-connected consumer.

As with any UI, there is of course plenty room for variation, customization and branding. Any such efforts must be accompanied, however, by careful consideration of their potential impact on the driver's attention and performance. Consider an iterative approach to this process, where potential new implementations or implementation changes are first prototyped on a desktop PC, tested in a driving simulator, and then refined one or more times before being incorporated into a production system and taken for a spin on the test track.

References

[1] Atkinson, C. and Shiffrin, M. (1968). Chapter: Human memory: A proposed system and its control processes. In Spence, K.W.; Spence, J.T. *The psychology of learning and motivation (Volume 2). New York: Academic Press*. pp. 89–1.

[2] Bach, K., Jæger, M., Skov, M. B., and Thomassen, N. (2008). You can touch, but you can't look: interacting with in-vehicle systems. In Proc. CHI 2008. ACM Press (2008), 1139-1148.

[3] Baddeley, A. (1994). The magical number seven: still magic after all these years? In: *Psychology Review 101* (2): 353–6.

[4] Baddeley A. (2003). Working memory: looking back and looking forward. In: *Nature Reviews Neuroscience 4 (10*): 829–839.

[5] 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*, University of Michigan Transportation Research Institute.

[6] Blattner, M., Sumikawa, D. and Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction* 4, 1 (1989), 11-44.

[7] Harbluk, L., Mitroi, S., and Burns, C. (2009). Three Navigation Systems with Three Tasks: Using the Lane-change test to Assess Distraction Demand. Proc. 5th Intl. Driving Symp. on Human Factors in Driver Assessment, Training and Vehicle Design.

[8] Hoffman, J., Lee, J., McGehee, D., Macias, M., and Gellatly, A. (2005). Visual Sampling of In-Vehicle Text Messages: Effects of Number of Lines, Page Presentation, and Message Control. *Journal of the Transportation Research Board* 1937, pp. 22-30.

[9] Hosking, S., Young, K., and Regan, M. (2007). The effects of text messaging on young novice driver performance. In: I.J. Faulks et al. (Eds.) Distracted driving, p.155-187. NSW: *Australasian College of Road Safety.*

[10] Kern, D., Mahr, A., Castronovo, S., and Müller, C. (2010). Making Use of Drivers' Glances onto the Screen for Explicit Gaze-Based Interaction. In Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI). Pittsburgh, PA, 11-12.

[11] Labský, M., Macek, T., Kleindienst, J., Quast, H., and Couvreur, C. (2011). In-car Dictation and Driver's Distraction: a Case Study. To appear in *Proceedings of the 14th International Conference on Human-Computer Interaction (HCII 2011)*. Orlando, FL, 9-14 July, 2011.

[12] Lamble, D., Kauranen T., Laakso M. and Summala, H. (1999). Cognitive load and detection thresholds in car following situations: safety implications for using mobile (cellular) telephones while driving, *Accident Analysis & Prevention*, Volume 31, Issue 6, pp. 617-623

[13] Mattes, S. (2003). The lane change task as a tool for driver distraction evaluation. In H. Strasser, H. Rausch & H. Bubb (Eds.), Quality of work and products in enterprises of the future. Stuttgart: Ergonomia Verlag.

[14] Muttart, J., Fisher, D. L., Knodler, M., and Pollatsek, A. (2007). Driving Simulator Evaluation of Driver Performance during Hands-Free Cell Phone Operation in a Work Zone: Driving without a Clue, *Transportation Research Record, 2018*, 9-14.

[15] Nuance Communications, Inc. (2011). T9 Write. http://www.nuance.com/for-business/by-product/t9-write/index.htm. Last accessed 18 May, 2011.

[16] Pettitt, M. A., Burnett, G. E., and Stevens, A. (2005). Defining driver distraction. Proc. *World Congress on Intelligent Transport Systems*.

[17] Rockwell, T. H. (1988). Spare visual capacity in driving – revisited: New empirical results for an old idea. In Gale, A.G., et al. Ed.: Vision in Vehicles II, Elsevier Science, North Holland, pp. 317-324.

[18] Stutts, J., Feaganes, J., Rodgman, E., Hamlett, C., Meadows, T., Reinfurt, D., Gish, K., Mercadante, M., and Staplin, L. (2003). Distractions in Everyday Driving, University of North Karolina at Chapel Hill, Highway Safety Research Center. S*tudy for AAA Foundation for Traffic Safety*.

[19] Suhm, B., Myers, B., and Waibel, A. (2001). Multimodal error correction for speech user interfaces. *ACM Trans. Comput.-Hum*. Interact. 8, 1 (March 2001), 60-98. DOI=10.1145/371127.371166 http://doi.acm.org/10.1145/371127.371166

[20] Tönnis, M. (2006). Time-Critical Supportive Augmented Reality – Issues on Cognitive Capture and Perceptual Tunneling. In *Proc. ISMAR 2006*, IEEE.

[21] Tsimhoni, O. and Green, P. (2001). Visual Demand of Driving and the Execution of Display-Intensive In-Vehicle Tasks. In *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting*.

[22] Tsimhoni, O., Smith, D., and Green, P. (2004). Address Entry While Driving: Speech Recognition versus a Touch-Screen Keyboard. In Human Factors, Vol. 46, No. 4. *Human Factors and Ergonomics Society*, pp. 600-610.

[23] Vredenburg, K., Isensee, S., and Righi, C. (2001). *User-Centered Design: An Integrated Approach.* Prentice Hall.

[24] Weinberg, G., Harsham, B., and Medenica, Z. (2011). “Investigating HUDs for the Presentation of Choice Lists in Car Navigation Systems.” To appear in *Proceedings of the 6th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design* (Driving Assessment 2011). Olympic Valley, CA, 27-30 June 2011.

[25] Wickens, D. (1984). Processing resources in attention. In: Parasuraman, R., Davies R., (Eds.), *Varieties of attention*, pp. 63-102. New York: Academic Press.

[26] Young, K.L., Regan, M., and Hammer, M. (2003). Driver Distraction A Review of the Literature. *Monash University Accident Research Centre*. Report no. 206.