AN INNOVATIVE VOCAL INTERFACE FOR AUTOMOTIVE INFORMATION SYSTEMS

Gennaro Costagliola, Sergio Di Martino, Filomena Ferrucci
Dipartimento di Matematica e Informatica, Università degli Studi di Salerno,via S. Allende, Baronissi, Italy

Giuseppe Oliviero, Umberto Montemurro, Alessandro Paliotti
Elasis S.C.p.A. - Veicolo SEE, Via ex-Aeroporto, Pomigliano D'Arco (NA), Italy

Keywords: HCI, Interface Usability, Mobile Computing, Spoken Dialogue

Abstract: The design of interfaces for automotive information systems is a critical task. In fact, in the vehicular domain the user is busy in the primary task of the driving, and any visual distraction inducted by the telematic systems can bring to serious consequences. Since road safety is paramount, it is needed to define new interaction metaphors, not affecting the driver's visual workload, such as auditory interfaces. In this paper we propose an innovative automotive auditory interaction paradigm, whose main goals are not to require visual attention, to be smart for expert users, as well as easy to use for inexperienced users. This is achieved by a new atomic dialogue paradigm, based on a *help-on-demand* mechanism, to provide a vocal support to users in trouble. Finally, we present some examples of dialogue based on such approach.

1 INTRODUCTION

In the last years, the in-car telematic systems, called also *Intelligent Transportation Systems* (ITS), achieved spectacular enhancements in their features. In fact, while the former systems were mainly focused on providing some route calculations, at the present more advanced commercial systems, like *Fiat Connect+* or *BMW iDrive* are becoming real in-vehicle computers, able to manage hundreds of features, such as the settings for the climate or the entertainment sections, the e-mail client, the GSM phone cell, the web browser, and so on.

Unfortunately, such growth in the number of offered services and information increased the workload inducted on the driver's visual channel, with negative consequences for the safety. This problem has a fundamental relevance in the automotive domain, where the user is normally busy in the demanding and mission-critical task of the driving. Thus if the system requires too much visual attention, it can distract the user from his/her main activity, with potentially fatal consequences.

As the road safety is the most important aspect when developing ITSs, it is now becoming clear that the next-generation of automotive applications will require large efforts for the definition of multimodal interfaces, intended both as complements or alternatives to the visual channel, and able to exploit the other user's sensorial channels (Gellatly, 1997). In particular, auditory interfaces can induct significant advantages, because the user can look at the road, and in the meanwhile interact with the system using the acoustic and/or the tactile channels. But in spite of such considerable advantages, a wide adoption of automotive vocal interfaces is currently limited both by strong industrial constraints on the in-car hardware, and both by some recognition problems due to the noisy car environment. Thus, the main challenge is to define a vocal interaction paradigm able to fully exploit the limited resource, that should be more effective than the traditional manual commands, be quick to use for expert user, be easy to use for novice users, encompass some error-recovery strategies, and overall, take in great care the road safety aspects.

The Fiat research centre "Elasis", and the Department of Mathematics and Informatics of the

Costagliola G., Di Martino S., Ferrucci F., Oliviero G., Montemurro U. and Paliotti A. (2004).
AN INNOVATIVE VOCAL INTERFACE FOR AUTOMOTIVE INFORMATION SYSTEMS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 9-14
DOI: 10.5220/0002653100090014

University of Salerno started a collaboration aimed to define an innovative, user-friendly interface for the next-generation of ITSs. The major purpose of this project was to keep in the highest priority the safety issues (thus minimize as much as possible the workload induced by the system), and in the meantime to propose something cost-effective to industrialize in the next two years.

In this paper we present the vocal interface resulting from that collaboration. In particular, we propose an interaction paradigm that is easy for novice users, as well as effective for experienced ones, taking particularly into account the management and the recovery from error situations.

The paper is organized as follows: in section 2 we introduce the main issues of HMI in the automotive field, in section 3 we describe the current proposal for vehicular speech interfaces, while in section 4 we illustrate the proposed approach, together with some examples of vocal interactions. Finally, in section 5 we present the conclusions and future work.

2 AUTOMOTIVE HUMAN MACHINE INTERACTION

"*The nation that develops and integrates an architecture that provides a seamless interface to the driver will dominate the automobile industry for many years to come*" (NTSC, 1997). This provision was stated by the US NSTC in 1997, to underline how much it is considered strategic and important to define interfaces for an effective automotive human-machine interaction. Today, seven years later, no one seemed able to fulfil such provision. In fact, while the ITSs are becoming even more some kind of traditional PC, able to connect to the WWW, check mail, play MP3 or DVD, the interaction with those systems is somehow far to be a well-established issue.

2.1 Interaction with ITSs

When dealing with telematic interfaces the main problem is that traditional HCI techniques and approaches, such as (Shneiderman, 1998) cannot be effectively applied. The main difference is that with desktop applications, designers can make the assumption that the user's attention is mainly focused on the interaction with the system. On the contrary, with ubiquitous computing (which encompasses the automotive scenario), designers cannot rely on a significant user attention, because usually in those domains the interaction with an informative system is only one task among the several actions achieved at the same time by the user. In particular, in the automotive domain, the user performs simultaneously a set of complex tasks. The main one is the driving, but concurrently (s)he can also perform a set of *secondary tasks*, involving interactions with entertainment systems, climate controllers, navigation aids, etc… Unfortunately, performing secondary tasks requires the allocation of some visual, manual and cognitive resources (Gellatly, 1997), implying the reduction of the attention devoted to the driving task, with an overall decreasing of the safety (Redelemeir, 1997). Thus the design and evaluation of ITS interfaces requires understanding not only the driver's interaction with the interface but also the effects of this interaction on driver performance, in order not to decrease the road safety. In particular, to avoid driver's cognitive overloads, the designer of an ITS interface has to make decisions not only about *what* information to show, but also about *how*, *where*, and *when* to show it. This new HCI approach is called *Driver-Centred* (Cellario, 2001). We can summarize these differences by arguing that while the main goal of a traditional desktop interfaces is to attract the visual attention of the user, to accomplish effectively his/her tasks, the main goal of ITSs interfaces is NOT to attract the visual attention of the user, to accomplish effectively his/her tasks. One of the ways to accomplish this task is to exploit the other human senses, such as the auditory one.

3 VOCAL INTERFACES

In literature there are dozens and dozens of works about auditory interaction, like (Cole, 1996) or (Shriver, 2000). Generalizing, we can say that all those efforts resulted in two main approaches for the definition of vocal interfaces: the one based on the *natural language*, and the one based on the *command word* (Westphal, 1999) (however some authors can use different terminology for this classification).

With the former approach, the systems should be able to "dialogue" with the users by using freely the natural language. This is a very effective, user-centered approach, because the system should be adapted to the man, and not vice versa. Unfortunately, this solution poses many practical problems, requiring a very complex Speech Recognition (SR) engine, that relies on a large amount of hardware resources and domain knowledge to effectively manage the possible user input. On the other hand, the command-word solution proposes the classical *computer-centric*

approach, were the user should learn how to use and interact with system. In fact, with this solution, in each state the system can accept only a relatively small vocabulary of words. If such set of words becomes larger, they typically must be arranged in a hierarchical structure, to maintain reduced the number of accepted word per state. This approach is much more simple and effective to implement, leading towards better result in the reconnaissance, but its main limitation is that the user must necessarily know the set of accepted words for each state. Hence, usually this approach requires some kind of support, which can be either vocal or visual. With the vocal one, the user listens a variety of options, like standard telephone-based call-center systems, while in the visual one the systems shows on a display the set of valid commands for each state.

3.1 Vocal Interfaces for ITSs

The definition of vocal interfaces in the automotive field according the driver-centered approach is a very interesting and opened research field. In fact, currently in literature does not exist any relevant theory or approach for the development of automotive dialogue-based interaction paradigms, but we can find a lot of generic guidelines like in (Rogers, 2000), together with many approaches about auditory support for the navigation aids, like in (Geutner, 1998). This because in the auditory domain, the automotive field introduces a wide set of new issues. In particular, the car interior is an acoustically hostile environment, encompassing a lot of ambient background noises, such as wind, climate fan, road conditions, speed, passengers, etc… Moreover the hardware resources should be limited, due to economical and industrial constraints, and the SR engine should be speaker independent, because either it is not supposable that the car's buyer spend a lot of time in training the system, and the vehicle can be used by a large variety of different drivers.

All these aspects lead towards a significant reduction of the system ability to recognize vocal commands. This means that for short-term ITSs, the developers have to discard the natural language approach, and to adopt the command word-like one, in order to obtain an effective rate of vocal interaction. But, as stated before, the command-word approach poses a series of problems for novice users. To this aim, the developers of ITSs vocal interfaces have to define an effective *dialogue paradigm*, able both to fit the imposed the industrial constraints, and to offer a satisfactory support for the users.

4 THE PROPOSAL

The aim of the collaboration between the research centre "Elasis" and the University of Salerno was to define a vocal interface, intended to manage the main secondary tasks for next-generation ITSs.

The main requirements for the system were:

- Easy to use for naïve user, by encompassing some kind of support
- Quick to use for expert users
- Easy and cost-effective to industrialize

Moreover, the system had to be implemented by using the ScanSoft Automotive ASR-1600 SR engine, with the limitation that the number of words should be about 30 for each state.

Thus, to define an innovative vocal interface, we had to specify (I) the system prompts, (II) the hierarchy of the command-words, (III) the error-recovery strategies, and (IV) the resulting interaction paradigm.

4.1 The hierarchy of commands

When dealing with the command-word approach, one of the main challenges is to deal with a limited vocabulary of accepted words, due to hardware constraints. Hence, we needed to define a meaningful hierarchy of the features we wanted to make accessible via vocal interaction. We grouped the words according to the modules of the ITS, leading to a multi-rooted tree, where the roots correspond to the modules of the ITS (i.e. the *Navigator*, the *Tuner*, the *CD*, the *Phone* and the *Services*), and the leafs represent the executable commands. To complete the hierarchy, we needed to introduce some items for navigating this tree from the roots to the leafs. We call *Non-Terminals* the commands representing the internal nodes of the hierarchy, and *Terminals* the command representing the leafs of the hierarchy. Obviously, the uttering of a terminal word leads to the execution of some action in the ITS, while a non-terminal word should be followed either by a terminal or non-terminal one.

Moreover, for an effective vocal interaction, the user should be provided with some features to move within the hierarchy, and to control the vocal interaction. To this aim we defined the following set of words, that can be used in every state of the system: the user can say "*Cancel*" to abort the voice recognition task, "*Undo*" to cancel the last accepted word (thus coming back of one level in the hierarchy), "*Repeat*" to make the system to say the last understood word, or "*Help*" to get the list of the words accepted in the current state.

Another fundamental aspect in the definition of the vocal paradigm is the relation between the

information shown by the GUI and the information accepted by the SR engine. Indeed, with current commercial ITSs, very often the vocal interfaces seems to be badly integrated with the underlying graphic interface. The result is that usually the vocal commands are not related at all with the information shown in the GUI. Our approach, instead, was to make the GUI and the SR to share the same context, i.e. changing of states operates via GUI reflects on the SR state and vice versa. This allows the users to mix visual/tactile and auditory inputs, making the interaction smarter and easier to learn.

4.2 The auditory prompts

In vocal interaction, the auditory prompts are, in some way, the basis of the "interface", because they drive the user through all the dialogue to achieve the intended task. Hence, it is fundamental that the prompts fit in with the ongoing dialogue (Krahmer, 1997), and that never be ambiguous, making the user always aware of the state of the vocal interaction.

When dealing with vocal interfaces, designers can exploit two main kinds of prompts: the "earcons" (Brewster, 1989) and the machine-driven dialog. In the former case, the system plays a tone to report an event of the interaction (such as the acceptance of a command or the completion of a task), while in the latter the system "says" one or more word, to the same aim. Each of the two approaches has its main advantages and disadvantages. The machine-driven one is very useful to guide novice users to their goals, because in each state the system lists the set of valid commands, but interactions result slowed by this forced iteration of options, most of which are presumably irrelevant to the user's goals. Instead, the earcons-based one leads towards a very quick interaction, but it can result hostile for novice users, who do not receive any kind of support.

In our proposal we mixed the two approaches: the interaction is mainly based on the earcons, and in particular any system output always terminates with and earcon, but, if the user appears to be in difficulty, the system starts to provide some kind of vocal support to the user. About the adopted earcons, we used a single earcon to represent the state when the system is able to manage a new input from the user, and a double earcon to highlight the successful end of a vocal interaction. This approach will be detailed in section 4.4.

4.3 The error-recovery strategies

In the automotive field, the SR engine very often has to deal with errors in the speech recognition. About the cause of these errors, we noticed that, during a vocal interaction with the system, two kind of fault situations can arise:

- The user does not utter any word.
- The word uttered by the user does not match any valid command.

In both cases, the system has to initiate some error-recovery strategy, but, in our opinion, these two situations imply two different kinds of problems. In the first case, most likely the user does not know what the accepted commands are for the specific state of the hierarchy, and thus an immediate support is required. The second situation, on the other hand, can be generated either from a wrong uttered word or from a system error in the recognition. These two fault situations requires different recovery strategies, but surprisingly, most of the current commercial systems manage the two conditions in the same way. Moreover, about the second situation, we were interested in understanding how much of the unmatched errors are caused by a user's error and how much by the system. To this aim, we conducted some evaluations on medium-high class cars on the market equipped with some of the most advanced commercial ITS. We found an average recognition rate of the 95% when dealing with the commands, and of almost the 92% when inputting numbers in the cell-phone dialing task. Notice that, even if those results seem a very good achievement, they mean that, when vocally composing a phone-number, there is a misunderstood cipher every eleven uttered! This quick survey lead us to an important consideration: if a recognition error occurs, it is very likely the system missed to recognise the word rather than the user uttered an invalid word. This consideration has some deep implications in the definition of error-recovery strategies. The most obvious is that if an error does occur, we do not have to "condemn" the user, with something like "*What you say is illegal*", but instead we should let the system take the responsibility of the error and focus on recovering. Moreover, in presence of a recognition error caused by the system, if it starts to list all the acceptable words, the only result is to annoy and irritate the user. Finally, it is widely recognized that one of the worst (and most irritating) approaches in presence of an error is to systematically request a repetition of the command to the user (Gellatly, 1997). This because, in presence of a recognition error, users act like as they would do when dealing with a human dialogue partner: they start speaking slowly, varying volume, pitch and rate of the pronunciation, as well as syllabicating the words. All these actions work well with human counterparts, but unfortunately they deteriorate very much the results of the SR engine.

Starting from these bases, we defined two different strategies for the error, having the main goal to keep the vocal interaction quick and effective. The main idea was to provide a contextual *help-on-demand* mechanism. This means that for each state the user can ask for some support, but the system does not provide any kind of vocal help until it is explicitly required by the user. However, because the user must be aware of the availability of a help, if the system detects a possible indecision in the user vocal interaction, it "suggests" to him/her the existence of the vocal support. The main issue for defining an effective auditory error-recovery strategy was thus to support the user only when (s)he really need for assistance, because too much (non demanded) help will unacceptably slow down the interaction, making the user discarding the vocal interface, with all the related security consequences. On the other hand, too little help will leave the user unable to continue the interaction, again making him/her not using the vocal system. Thus, our main problem was to understand *when* the user is in trouble and needs for assistance. The approach we chose is to select the strategy to adopt based on the happened error:

- If the system does not understand the uttered word, we made the assumption that likely the system is in wrong and thus it should not start with providing any support, but just asks to the user to repeat the word. Then, only if there is a second mismatch, this probably means the user is in wrong. In this situation, the system proposes to the user to say "help" to get the contextual assistance, i.e. a list of all the commands accepted in that particular state.
- If the user does not utter any word after the prompt, very likely it means (s)he does not know what to say, and then (s)he immediately requires some kind of support. Again, in this case the system proposes the user to say "help" to get the contextual assistance.

With this approach, we are able to discriminate when the user really has to be supported, thus making the whole vocal interaction more agile and effective.

4.4 The proposed paradigm

The resulting vocal interaction paradigm is derived by all the concepts exposed above. and is depicted in Figure 1, by using the statechart formalism. To start a vocal interaction with the ITS, the driver should press a button located on the steering wheel, and named Voice Recognizer (VR). As a result, the system activates the on-board microphone and the SR engine, and plays a "ready earcon", meaning that it is ready to manage a vocal input. In every state of the vocal dialogue, then, three conditions may occur:

1) The user utters a word that is matched by the system. In this case, the system goes in the "Matched" State. If the word is a terminal command, then the system executes it, otherwise, the system moves in the hierarchy, goes back to the "Start" state, and plays again a "ready-earcon", meaning that the word was matched and the user can utter the next word.
2) The user utters a word that is not matched by the system. In this case, the system goes to the state "Unmatched". At the same time it prompts the problem to the user by saying "Sorry?" and playing the "ready-earcon". Notice that in this situation no explicit vocal help is provided. Now again the three conditions can happen:
   - If the new inputted word matches a valid command, than the system goes to the "Matched" state, proceeding as condition 1.
   - If the new inputted word does not match a valid command, this very likely means that the user is in wrong. Thus the system says *"Unable to match. If you need assistance please say **help**, otherwise input a choice"*, goes back to the "Unmatched" state and plays the ready earcon.
   - If the user utters nothing the system aborts the SR task.
3) The user does not utter any word for more than 5 seconds after the ready earcon. In this case, the system goes says *"If you need assistance please say **help**, otherwise input a choice"*, and then acts as in the previous situation.

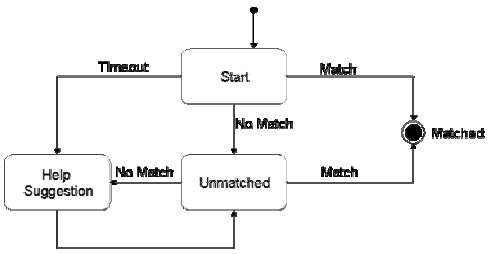

Figure 1: The proposed vocal interaction paradigm

4.5 Some examples of dialogue

In the following we provide some examples of auditory interaction, to exemplify the vocal paradigm proposed above.

In the first example, let us suppose the user wants to listen to the track number 3 of the CD when the navigator module is activated:

User: *(click on the VR button)*
System: *(ready-earcon)*
User: *"CD"*

System: *(ready earcon)*
User: *"Track"*
System: *(ready earcon)*
User: *"Three"*
System: *(ok earcon and play track 3)*

Because the vocal engine is aware of the active module, to achieve the previous task when the CD module is active yet, the user can avoid to say "CD", to speed-up the interaction.

In the next example it will be shown how the system manages a double recognition error. Let us suppose again the user wants to see the entire path calculated by the navigator on the display.

User: *(click on the VR button)*
System: *(ready-earcon)*
User: *"Navigator"*
System: *(ready earcon)*
User: *"Visualize"*
System: *(mismatch) "Sorry?"*
User: *"Visualize"*
System: *(mismatch) "Unable to match. If you need assistance please say* ***help****, otherwise input a choice" (ready-earcon)*
User: *"Show"*
System: *(ready earcon)*
User: *"Path"*
System: *(ok earcon and show the path)*

Finally in the following example we show how the system manages a timeout error. Let us suppose the user wants to redial the last number on the cell.

User: *(click on the VR button)*
System: *(ready-earcon)*
User: *"Phone"*
System: *(ready earcon)*
User: *(silence for more than 5 seconds*)
System: *"If you need assistance, please say* ***help****, otherwise input a choice" (ready-earcon)*
User: *"Help"*
System: *"You can say: Dial, Call contact, or Redial. Please input a choice". (ready-earcon)*
User: *"Redial"*
System: *(ok earcon and redial last number)*

5 CONCLUSIONS

Safety on the roads is one of the main goals for everyone involved in the automotive field. The advent of in-car ITS systems based on a visual interaction can distract the user from the main task of driving the car, with potentially fatal effects. On the other hand, the availability of even more complex telematics requires even more complex and advanced interaction mechanisms.

In this paper we presented a work developed jointly by the research centre "Elasis" and the University of Salerno, aimed at defining a novel approach for the automotive vocal interfaces. The proposal, based on the *command word* paradigm to match the hardware constraints, encompasses a new atomic dialogue paradigm, based on earcons and a *help-on-demand* mechanism, to provide a vocal support to users in trouble. The result is a smart auditory interface for expert users, but also user-friendly. Finally, it is important to underline that by using this paradigm, the user can interact with the telematic system by using exclusively the auditory and the tactile channels, thus without distracting the visual attention from the road.

REFERENCES

Brewster S, Wright P. and Edwards A., An evaluation of earcons for use in auditory human-computer interfaces. *Proceedings of InterCHI'93*, Amsterdam, 1993

Cellario M., Human-Centered Intelligent Vehicles: Toward Multimodal Interface Integration. *IEEE Intelligent systems*. 2001

Cole R. et al., *Survey of the state of the art in human language technology*. Cambridge University Press, 1996.

Gellatly A., *The use of speech recognition technology in automotive applications*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

Geutner et Al., Conversational Speech systems for on-board car navigation and assistance. *ICSLP '98*, Adelaide, Australia, 1998

Krahmer E., Landsbergen J., Pouteau X.. How to Obey the 7 Commandments for Spoken Dialogue Systems. Proc. *(E)ACL workshop on Interactive Spoken Dialog Systems*, J. Hischberg, C. Kamm & M. Walker, Madrid, 1997

NTSC HCTS Team, The Human Centered Transportation System of the Future. *Proc. ITS America 7th Ann. Meeting*, Washington D.C., 1997.

Redelemeir D.A., Associations between Cellular-Telephone Calls and Motor Vehicle Collisions. The *New England Journal of Medicine*, Vol. 336, February 1997.

Rogers S. et Al., Adaptive User Interfaces for Automotive Environments, Proc. *IEEE Intelligent Vehicles Symposium*, 2000.

Shneiderman B, *Designing the User Interfaces*, Addison Wesley, 1998

Shriver S., Rosenfeld R., Keyword Selection, and the Universal Speech Interface Project. *Proc. AVIOS 2002*, San Jose, California, 2002

Westphal, M., Waibel,A., Towards Spontaneous Speech Recognition For On-Board Car Navigation And Information Systems. *Proc. of the Eurospeech 99*, 1999.