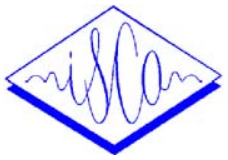

EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

Robust Spoken Dialogue Management for Driver Information Systems

Xavier Pouteau, Emiel Krahmer, Jan Landsbergen
E-mail: {pouteau/krahmer/landsbrn}@ipo.tue.nl
IPO, Center for Research on User-System Interaction
P.O. Box 513, 5612 MB Eindhoven, The Netherlands

Abstract

Considering the limitations of Speech Recognition for the development of user-system dialogues in real applications, robustness is a primary objective. In this paper, we describe the most essential characteristics of the Dialogue Manager of a driver information system that is controlled by voice, mainly showing how its design has been driven by the characteristics of voice in such a dialogue. We present the main methods used by the Dialogue Manager to come to an effective balance between robustness and efficiency. We illustrate them with examples from the first implementation of the system.

10.21437/Eurospeech.1997-582

1 Introduction

In recent years, there has been a growing tendency to include speech as an in- and output mode for user interfaces. This is due to the expectation that including speech broadens the 'bandwidth' of interaction between user and system, and allows for a more 'natural' communication. To translate these expected advantages into actual characteristics of the intended system, a number of problems associated with robustness of speech input have to be addressed [Lea 1994], [Leiser 1993]. Even though current *speech recognition* (SR) techniques perform well, it is also commonly accepted that errors can never be precluded. This entails that the *Dialogue Manager* (DM) of a vocal interface has to include a method which *deals with* errors.

2 Application area: driver information system

Driver information systems (generally containing HiFi equipment, but recently also navigation computers, traffic messaging and mobile telephone) are increasingly complex, and one could argue that the only way to operate them while driving is by voice. But, since the driving activity should always have the highest priority for safety reasons, a vocal interface for such a driver information system has to satisfy certain strong requirements: it should avoid irritating the user, mainly by being very reliable and not obtrusive. An obvious complicating factor is that the car is an acoustically hostile environment, which puts an additional burden on the speech technology. The challenge to design and implement an acceptable voice-operated driver information system is taken up in the LE-2277 (VODIS) project [Arévalo 1995]. The general architecture of the VODIS system is given in figure 1. On the one hand, the VODIS project aims at developing the relevant techniques for robust SR. Fortunately, the kind of noise in the car (wind, tires, engine) is to a certain degree predictable from the available speed data, and this gives several reference points for effective noise reduction. On the other hand, VODIS also has an active policy to *handle* errors. This is part of the task of the Dialogue Management module (DM).

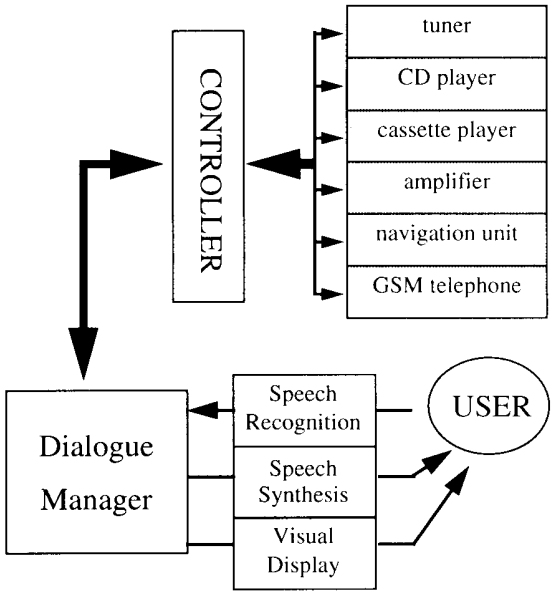

figure 1: general architecture of the VODIS system

3 Requirements for a robust Dialogue Manager

The DM in VODIS should control the interaction between the user and the actual driver information system (Robert Bosch GmbH's BERLIN RCM303A). In this sense, the DM is the central module of the system; it functionally controls all the parts of the system. This is reflected in the following tasks of the DM:

i interpreting the user's spoken input,

ii controlling the results of SR, by running recovery strategies in case of a SR error,

iii handling the dialogue with the user by taking the initiative in some cases, and leaving the initiative to the user in other cases. As illustrated in [Cahour 1989], [Fraser&Thornton 1995], the level of initiative has to correspond with the competence of the partners in the dialogue,

iv controlling the devices, by performing the corresponding changes of state requested by the user through the

Controller (which is a software interface to operate the hardware devices),

v selecting the appropriate sub-grammars in the language model of the SR unit, in order to limit the size of the active grammar, as a means of 'on-line' perplexity reduction.

3.1 Limitations concerning the spoken input

As the content of each item in the list above shows, robustness is a relevant issue. Consider issue i: interpreting the user's input. For every input of the user, the SR unit returns an n-best list to the DM. Notice that the DM cannot determine *a priori* whether this n-best list:

(a) contains the right candidate (the one that corresponds with the user's input),

(b) does not contain the right candidate because of a user's mistake (e.g., utterance out of the SR grammar),

(c) does not contain the right candidate because of an error in the SR unit.

Though it is expected that (a) will occur in most cases, cases (b) and (c) should also be envisaged at the design stage because they will also occasionally occur. Considering both partners of the dialogue, methods that can be developed to solve problems in speech recognition belong to two categories:

1 the ones relying on the system, mainly filtering methods (on the basis of semantic information and of the context of the dialogue, so as to reject the *non-plausible* candidates of the n-best list),

2 the ones relying on the user, by letting him/her intervene in the dialogue to indicate the correctness of the results of the speech recognition process.

It has to be noted that 1 is at best a partial strategy, and hence does not ensure complete robustness for several reasons: in most dialogue states, there will not be a single, most plausible candidate, and moreover, there is no way to guarantee that the right candidate (i.e., the user's actual utterance) is an element of the n-best list to begin with. On the other hand, using 2 systematically is not a good choice either, because this would not only slow down the dialogue, but also increase the chances of SR errors by increasing the number of utterances to treat before completing one task. Thus, to deal with the limitations of SR, relying on the user is mandatory, but the strategy of the Dialogue Manager has to avoid putting too much burden on the user by explicitly requesting his/her intervention.

3.2 Design issues

To achieve a compromise between a robust and a not too intrusive system, several design choices have been made for the DM (see also [Krahmer *et al* 1997]). Designing the feedback is a core issue from this perspective. Feedback messages have been defined in such a way that the user always gets clues about what has been recognized, in a context-dependent way.[1] The following rules have been applied:

- a minimal level of feedback always has to be provided. This is not only a matter of robustness, but it also has great influence on the acceptability of the system (e.g., [Nguyen-Xuan & Hoc 1987]),
- obviously, both *form* and *content* of the feedback have to be taken special care of, because they have direct influence on the user's reaction to the feedback message: a syntactically marked yes/no question or a clear question contour (H*HH% in the terminology of [Pierrehumbert & Hirschberg 1990]) will cause the user to feel forced to explicitly confirm or reject,
- last but not least, for reversible tasks (through an available'undo' command), and in the case of high confidence scores from the SR, the feedback implicitly consists in achieving the task, together with the acknowledgement by the system that the task has been achieved. The backbone of this strategy is a guideline which says that every input from the user should be handled as deep as possible, thus lowering the number of interventions that is required to achieve a task. This also helps to avoid distracting the user's attention from the traffic environment.

4 Description of the Dialogue Manager in VODIS

4.1 Objectives of the project

The VODIS project aims at developing two versions of the user interface: a keyword-based interface in a first stage (the user communicates through short utterances based on command keywords) and a spontaneous spoken input interface in a second stage. The description of the system hereinafter corresponds with the first version of the system. It will be evaluated at the end of 1997 and the evaluations will serve as a basis to refine the design of the second phase.

4.2 Content of the Dialogue Manager

A turn in the dialogue consists of the following steps (see also figure 2):

- generation of forests (part-of-speech tagging). Every word of every candidate out of the n-best list is associated with its possible lexical descriptions (meaning that a word having several lexical entries in the "Vocabulary lexicon" will cause several forests to be generated),
- parsing of every forest generated for each candidate. Due to the simplicity of the language (command keyword-based utterances), no syntactic parsing is needed and this step only consists in a semantic parsing, according to the description of the semantic content provided by the "Task lexicon",[2]
- the alternative lists of possible 'Task Units' are passed to the Dialogue Controller, which orders the candi-

[1] The DM can provide feedback to the user via synthesized speech and via a display. The visual feedback contains no information which is not also conveyed by speech; it merely serves to compensate for the transience of speech. In this paper we focus on spoken feedback.

[2] this is not the case for the second prototype, in which more elaborated NLP techniques ([Ward 1994]) will be used.

dates according to the context of the application and the history of the dialogue (see example 5.1 below),

- the top candidate of the list is pushed on the top of the "Dialogue Stack", and this candidate is further processed in the dialogue with the user,
- processing the Task Unit (see below in section 4.3) leads to a new *state* of the context (defined by the content of the Stack), and the vocabulary available in this state is notified to the SR unit (selection of the "SR grammar") to support the recognition process of the user's next input.

For the current version of the vocabulary (which will be updated according to the results of the evaluations), there is no ambiguity in the process of tagging and parsing, meaning that for each candidate out of the *n*-best list (n has a maximal value of 5), there is only one Task Unit passed to the Dialogue Controller.

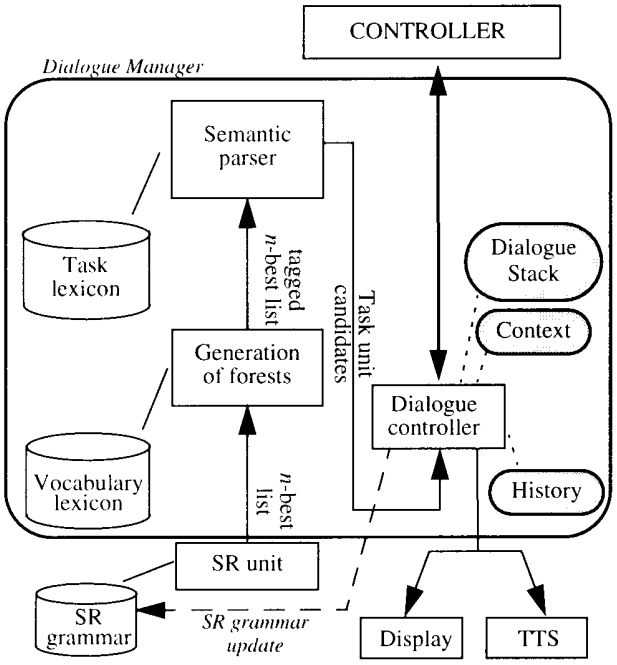

figure 2: software architecture of the DM in VODIS

4.3 Dynamic characteristics

The DM should avoid distracting the user from his main task: driving a car. This means that a compromise has to be found between efficiency and robustness. In this paragraph, we describe the main aspects of the DM which aim to achieve this.

4.3 .1 Validation stage

The preconditions of the Task Unit pushed on the top of the Stack are first evaluated according to the context (which is a local representation of the state of the application, that is, the devices controlled by the user —see figure 1). If the preconditions are not met, the user is made aware of the reason why the requested operation is not possible. Otherwise, the Task Unit is further processed, by providing an immediate feedback to the user[3], or sending the message to the controller so as to trigger the change of state corresponding to the user's input (in the latter case, the operation can be undone).

Once the feedback has been provided to the user, the Task Unit on the top of the Stack (not yet validated) has to be validated, and the validation (or rejection) would then also apply to the results of the speech recognition. The user can then reject the proposition of the Dialogue Manager (by saying "no"), or validate:

- by explicit confirmation (e.g., "yes", "OK"),
- by uttering a follow-up command (which forms an implicit agreement with the content of the feedback),
- by keeping silent (the non-rejection being interpreted as an implicit confirmation).

4.3 .2 DM initiative

So far in this paper, we have considered the reaction of the system to a user's input, that is, a *user-initiated* dialogue. This situation corresponds to the case where *the user knows what to do* and *how to do it*. At this stage, we also have to take into account that there will have to be a training period, which corresponds to the fact that *the user is not a professional operator* of the system but a consumer, and will have to learn how to operate the system. In VODIS, this is done in a simple way: since the application itself (the set of controlled devices) can be described as a state machine, any state corresponds with a set of possible operations (see example 5.2 below). So at any stage in the dialogue, the user can ask for assistance, which will cause the system to provide the following pieces of information:

i the current state of the system,

ii the possible operations, together with the way to express the corresponding commands.

We expect this functionality of the Dialogue Manager to improve the learnability of the system, as suggested by the experiment described in [de Vriest & Johnson 1997]. In addition to this user-initiated help, the system can also provide those messages automatically, according to the following scheme: when the system enters a new state on a user's request, and when that state is an intermediate one along the sequence of Tasks Units that complete a task, then it is obvious that additional Tasks Units have to be completed. The mechanism implemented in VODIS (following [Cosky *et al* 1995]) consists in providing the user with the help message in case no dialogue would take place after the transition to the intermediate state.

[3] As already mentioned, feedback is a highly important issue to which a great deal of attention has been paid during the design of the system. The feedback corresponding to each task has been individually defined according to the desired characteristics of being informative and non-intrusive. In case the action is 'triggered' directly, a complementary feedback is provided if the perceivable effect of the operation is not obvious enough.

5 Examples

The examples below give better insight in two of the mechanisms mentioned earlier in this paper. The VODIS system is actually developed in a French and a German version, but examples are given here in English to improve readability. The user's utterances are prefixed with a "U", and the interventions of the system with an "S". The *n*-best lists returned by the SR unit to the DM are prefixed with the symbol SR. Spoken interventions are between «», other operations (i.e., system operations) are mentioned between // \\.

5.1 Example1: recovering from an SR error

The user wants to make a phone call. A phonebook contains the details of addressees under user-chosen nicknames. Let's suppose that 3 of the nicknames are "Bill", "Jill" and "Phil".

<U1>: «call Phil» (SR: "call Bill", "call Jill", "call Phil").

<S2>: «calling Bill»

The system gives immediate feedback on the first candidate

<U3>: «no»

<S4>: «calling Jill»

since first candidate is challenged, it is discarded, and the second one is considered

<U5>: «cancel»

the user resets the SR results and re-utters the command

<U6> «call Phil» (SR: "call Bill", "call Jill", "call Phil").

since it is a repair dialogue, initiated through U5, the DM takes into account the candidates already challenged (in the history of the dialogue, see figure 2).

<S7>: «Calling Phil»

<U8>: «OK»

<S9>: // the system dials the corresponding number \\

5.2 Example2: providing spoken help

We take here the example of a less experienced user, and illustrate the mechanism issued in paragraph 4.3 .2.

<U1>: «select navigation» (SR:" select navigation")

<S2>: «Switching to navigation»

user keeps silent, which is considered as an implicit confirmation

<S3>: // the system selects the navigation unit \\

the user keeps silent: the system proposes the possible operations

<S4>: «In navigation mode, you can select:

enter destination, to select a destination for route guidance,

show actual position's map, to see the current position of the vehicle on the map,

select side information, to get additional information on the route»

reminds the current state of the system, and then proposes the phrasings corresponding to the different possible operations

<U4>: «enter destination»

...

6 Conclusions

When developing an in-car spoken dialogue system the limitations of SR have to be compensated for in a robust, but not intrusive manner. The main techniques have been described at various levels of detail in this paper. Furthermore, in a spoken dialogue, and especially for consumer application, the performances of SR units are not the only factor that influence the efficiency of a dialogue: user's mistakes, due to a wrong representation of the system have dramatic impact. This is why much attention has been paid to the design of the feedback, as not only a way to avoid giving the user an erroneous impression of the system, but also to improve learnability. The adequateness of these techniques will be a core issue in the evaluation phase, and we expect the results of the evaluation to give clues to improve the design of the current version of the system in a first stage, and to indicate the aspects of the dialogue that a spontaneous spoken input interface has to focus on, for the second version of the system.

Acknowledgements

This work was funded by the European Committee as part of the Language Engineering/Telematics Applications Program, project LE-1 2277 (VODIS).

7 References

Arévalo, L., 1995, "VODIS: Advanced Speech Technologies for Voice Operated Driver Information Systems", Technical Annex

Cahour, B., 1989, "Niveau de compétence des interlocuteurs et répartition de l'initiative dans les dialogues de consultation", INRIA, Research Report N°1124

Cosky, M., et al., 1995, "Talking to machines today and tomorrow", in Designing for the user, *AT&T Technical Journal*, 81-90.

Fraser, N. M., Thornton, J. H., 1995, "VOCALIST: a Robust, Portable Spoken Language Dialogue System for Telephone Applications", in Eurospeech'95 Proceedings, 1947-1950.

Krahmer, E., Landsbergen, J., Pouteau X., 1997, "How to obey the 7 commandments for spoken dialogue", in Proc. ACL/EACL *Workshop on Spoken Dialog Systems*, July 11-12, 1997, Madrid Spain.

Lea, W., 1994, "Developing usable voice interfaces", in Journal of the American Voice I/O Society, 16.

Leiser, R., 1993, "Driver-vehicle interface: dialogue design for voice input", in: Driving future vehicles, A. Parkes, S. Franzen (eds.) Taylor & Francis, 275-294

Nguyen-Xuan, A., Hoc, J.-M., 1987 "Learning how to use a command device", in *Cahiers de Psychologie Cognitive*, 7, 1, 561-572.

Pierrehumbert, J., Hirschberg, J., 1990, "The meaning of intonational contours in the interpretation of discourse", in P. Cohen et al. (eds), *Intentions in communication*, 271-311, MIT Press.

Vriest, G., de, Johnson, G., J., 1997, "Spoken help for a car stereo: an explanatory study", in *Behavior & Information Technology*, 16, 2, Taylor & Francis, 79-87.

Ward, W., 1994, "Extracting information in spontaneous speech", in Proc. ICSLP 94, Yokohama, 83-86