Enhancing the Usability of Multimodal Virtual Co-drivers

Niels Ole Bernsen and Laila Dybkjær
NISLab, University of Southern Denmark, Odense
nob@nis.sdu.dk, laila@nis.sdu.dk

Abstract: This chapter discusses a series of four user-oriented design analysis problems in a research prototype multimodal spoken language dialogue system for supporting drivers whilst driving. The problems are: (a) when should the system (not) listen to the speech and non-speech acoustics in the car; (b) how to make use of the in-car display in conjunction with spoken driver-system dialogue; (c) how to identify the present driver as a basis for building user models of the driver; and (d) how to create useful on-line adaptive user modelling of the driver.

Key words: in-car spoken dialogue systems, in-car multimodal interaction, driver identification, in-car adaptive user modelling

1. INTRODUCTION

Spoken language dialogue systems (SLDSs) are now firmly positioned in the market and appear set to become available in an increasing number of languages and for a rapidly increasing number of tasks. Current commercial SLDSs help people solve a single task or sometimes several independent tasks through spoken dialogue. The dialogue is still mostly being conducted over the phone but open microphone applications are beginning to proliferate as well. The tasks solved are mainly information retrieval and/or information entry tasks but also in this respect the field is rapidly diversifying into reservation tasks, training tasks, tasks involving important elements of negotiation between user and system, etc.

A typical commercial SLDS has speaker independent speech recognition; up to several thousand words in its vocabulary; modest natural language processing of the recogniser's output; increasingly modular dialogue management which often interacts with a domain database; no natural language generation; and spoken output production by means of concatenated speech, i.e. through on-line combination of recorded sentences, phrases, and words. However, due to its superior flexibility and reflecting recent increases in perceived quality and immediate intelligibility, speech synthesis is now also starting to be used for certain languages. In order to maintain control of users' spoken input behaviour and also to support the user in actually carrying out his task, the dialogue is mostly system-directed, especially in systems intended for irregular and infrequent use. The system thus "takes the user through the task" to its completion in a structured, more or less flexible fashion [Bernsen et al. 1998].

Meanwhile, next-generation systems are gathering in the pipeline based on progress in research. These SLDSs will represent solutions to a series of next-step technical challenges, including robust speech recognition in noisy conditions; very large vocabulary speaker-independent speech recognition and understanding; recognition of the pronunciation variants produced by native speakers of different languages; confidence-adaptive dialogue through dialogue manager use of speech recogniser and natural language processing confidence scores; natural language processing of fully spontaneous spoken input, so that the SLDS no longer has to conduct system-directed dialogue when mixed-initiative dialogue or (still task-oriented) conversational dialogue is more appropriate; dialogue management of mutually dependent tasks; integration of adaptive user models built on-line from observations of a particular user's behaviour; situation-aware system dialogue based on knowledge of the current situation of the user-system complex; and template-based natural language generation. Moreover, next-generation SLDSs will probably no longer be mainly speech-only, or unimodal, systems but will increasingly combine speech with other modalities for information representation and exchange, enabling multimodal dialogue. Also, systems will increasingly migrate to mobile environments and devices, making location-awareness and mobile internet access highly desirable system properties in many applications.

This chapter discusses a cluster of user-oriented design analysis problems in a research prototype multimodal SLDS which addresses the next-step challenges mentioned above in the context of supporting car drivers whilst driving. Common to the problems discussed is that they concern the design of important aspects of user-system interaction for in-car environments, that they presented key usability challenges during the development of the research prototype, that the problems appear to be unsolved so far, and that

finding solutions to them may be of interest to designers of mobile multimodal spoken language dialogue systems more generally. The research prototype system is called VICO (Virtual Intelligent CO-driver) and is being developed in the European HLT VICO project which began in March 2001 and has a duration of three years. The project partners are Robert Bosch GmbH, DaimlerChrysler AG, Istituto Trentino di Cultura, Phonetic Topographics N. V., and NISLab. NISLab is the developer of natural language understanding, dialogue management and response generation components in three languages: English, German and Italian for the VICO prototype discussed in this chapter.

In the following, Section 2 provides a general description of VICO functionality and architecture. Sections 3 through 6 discuss the following problems: when should VICO listen? (Section 3), why use multimodal speech-graphics output in the car? (Section 4), how to identify the driver? (Section 5), and which aspects of the driver's behaviour should VICO model on-line? (Section 6). Section 7 concludes the chapter by discussing some of the issues for which additional research seems clearly needed.

2. THE VICO SYSTEM

The car driver's environment is both a challenge and an opportunity for next-generation SLDSs developers. Important challenges include noise, from the car itself (engine, air flow, tyres, in-car climate regulation, etc.), traffic, rain, passengers, and in-car entertainment systems; very large vocabulary recognition, such as of +100.000 names of German regions, cities, streets, etc.; traffic safety; and ease of use by a large and heterogeneous user population. The opportunities are equally important. Car driving is a safety-critical, heads-up, hands-occupied activity in which the driver is mostly free to speak to fellow passengers and equipment but can only to a very limited extent expend valuable attention resources on GUI (graphical user interface) devices, such as screens, hand-held remote controllers, or keyboards. The car industry and user need studies concur that navigation is the "killer application" task for in-car SLDSs but that spoken interaction might be useful for many other tasks as well [Manstetten et al. 2002]. Moreover, there are strong indications that spoken car navigation and use of speech in the car more generally, cannot useably be realised by command-based SLDSs [Minker et al. 2002, Salmen 2002]. The reason is that drivers are not able to remember the required, increasingly large number of spoken commands needed to operate in-car SLDSs. For reasons such as the above, the development of a usable and versatile in-car SLDS is an obvious "technology push" challenge whose "user pull" can be taken for granted.

To address this challenge, we have built the first of two planned prototypes of a natural interactive, multilingual (UK English, German, and Italian), cross-lingual (recognising accented pronunciations of proper names), confidence-adaptive, and multimodal in-car spoken dialogue system. The first prototype enables navigation assistance, including streets and street numbers, parts of cities, cities, and, when relevant, parts of country for the Trentino Province in Italy; navigation to 25 different point of interest types in this area, such as cinemas, petrol stations, doctors, and airports; information about the VICO system itself (UK English only); and hotel reservation over the internet (simulated) based on a number of driver-defined hotel selection constraints and followed by the actual hotel reservation. The first prototype also includes an observation-based user modelling module which enables VICO to adapt its dialogue behaviour to the current driver's hotel preferences. Finally, restaurant reservation is enabled in the hotel in which the user has booked a room.

The second prototype will provide additional user modelling functionality based on on-line gathered data on particular drivers as a basis for adaptive system behaviour; navigation to addresses and points of interest in Germany and Greater London; real hotel reservation over the internet; scenic route planning including web-based information on touristic points of interest, such as castles and churches, which will be accessed using GPS-based location awareness; car manual information; news reading; and spoken operation of in-car devices. Throughout its interaction with the driver, VICO will maintain some amount of situation awareness with respect to the car, for instance by avoiding intrusion on the driver in dangerous traffic situations. The driver-VICO dialogue is spontaneous natural interactive dialogue, allowing the driver to address any task and sub-task in any order and using any appropriate linguistic form of expression. Finally, taking into account the in-car and out-of-car (traffic) environment, VICO will incorporate aspects of multimodal communication. Thus, VICO will be activated by pushing a button on the steering wheel and the system will provide both spoken output and graphics display output.

In the following sections, we describe our approach to four key usability challenges facing VICO interaction design and development.

3. VICO HAPTICS: HOW AND WHEN TO MAKE VICO LISTEN?

An in-car spoken dialogue system faces the problem of figuring out when the registered acoustics in the cabin is actually input meant for the system or just background noise. One of the really hard problems arises from potential

cross-talk between the driver and the passengers while the system is listening.

Ideally, start-stop control of recognition should be performed without being noticed by the user [Furui 2003]. However, given current recognition technology and in order to reduce the amount of recognition problems and nonsense dialogues which may arise from driver-passenger cross-talk, limitation should be imposed on the periods during which the recogniser is listening. In the VICO project it has been decided to introduce a push-to-activate button for this purpose. To start VICO and make the system listen, the user must push the button.

3.1 Button design and interaction

The design of the push-to-activate (PTA) button has not been finally decided yet. However, it seems likely that the button will be positioned on the steering wheel. The button will be red when the recogniser is inactive and green when the recogniser is active. If the button is red and the user pushes it, it will turn green as soon as the recogniser is ready.

In addition, we will experiment with acoustic awareness so that the user will not have to look at the button to see whether it is actually red or green. Acoustic awareness may be stimulated through non-speech sound or through spoken words or phrases, such as “hello”, or “good morning”. We expect that a non-speech sound will be felt less intrusive during daily use of the system compared to using words or phrases to indicate that the system is ready. When the recogniser goes inactive after a period of input inactivity (see below), this may be indicated through non-speech sound as well in addition to the button turning red. Using speech for this purpose, such as saying “bye”, would seem less appropriate since the system may still be talking to the driver about the task, for instance by continuing to provide navigation instructions after the recogniser has turned inactive.

The need for some kind of acoustic feedback on when the system is listening is supported by a set of Wizard-of-Oz experiments (see also Section 4). In those experiments, we only used a “button” on a display. The user was not supposed to push anything. The “button” would turn green when the system was ready to listen. However, users were not always aware of the state of the button because they were occupied driving the car and thus were not sure when they were supposed to start speaking. Although there is a difference between just passively waiting for the button to turn green and actively pushing a button which is then expected to become green soon thereafter, it still seems likely that acoustic feedback will be appreciated since it relieves the driver from having to keep an eye on the red/green

colour of the button before speaking. The acoustics is sufficient to tell the driver when the recogniser is open and when it has closed.

3.2 When to turn off the recogniser

Since we have decided that the recogniser will not just remain open once the PTA button has been pushed, we also have to find out when it is appropriate to turn off the recogniser. Clearly, it would be unacceptable that the driver has to push the button each time s/he wants to say something during an ongoing dialogue with VICO. On the other hand, however, the longer the recogniser remains open, the larger becomes the risk that it attempts to recognise speech not meant for the system, such as driver-passenger cross-talk. We have identified the following cases in which it seems appropriate to turn off the recogniser by means of a timeout function:

– a task has been completed and the driver does not initiate a new task within the following, say, 8-20 seconds;
– a driver stops interaction in the middle of a task and does not provide input for 8-20 seconds;
– a driver cancels an ongoing task and does not provide new input for 8-20 seconds.

A task is considered “completed” when the negotiation with the user is finished. A user may, e.g., have asked for route guidance to a particular address or point of interest. Once the system and the user have agreed where to go, the task is “completed” although the system may continue to provide (output-only) route guidance for the next 100 kilometres or more.

The system stacks a non-completed task in case the user wants to return to the task in order to complete it. For instance, a traffic situation may occupy the driver’s attention for more than 8-20 seconds, in which case the recogniser closes down. The system must be able to easily restore the dialogue state when the user pushes the button anew in order to continue the unfinished dialogue.

Clearly, the solution just proposed does not completely remove the background noise problems caused by driver-passenger cross-talk and other hard-to-model noise factors. In particular, the driver may still be talking to passengers while at the same time trying to have a dialogue with VICO. We do not have data that tells us how often this will be a problem. The system might try to diagnose the problem, when it occurs, through out-of-vocabulary word modelling, confidence score analysis, and other means. Thus, measures to identify input which was not meant for VICO may have to be taken by system modules other than the recogniser. However, this still leaves open the question of how to deal with the problem, when diagnosed. In one approach, VICO simply ignores cross-talk input. In another, VICO

applies its user modelling capabilities to remind frequent cross-talkers that driver-passenger cross-talk is counterproductive to getting the task done through spoken interaction with the system (cf. Section 6).

4. VICO GRAPHICS: WHEN MIGHT THE DRIVER LOOK?

Existing text-and-pointing input car navigation systems include a display on which output to the driver is shown throughout interaction. In particular, the display provides feedback on the driver's input when spelling addresses. The display may be small and without map information, using an arrow to show in which direction to turn next, or it may be somewhat larger and display a map showing the present location, direction, and planned route of the car in addition to the text and iconic information which is available on the small display. Navigation information on the screen is accompanied by spoken instructions on where to turn left or right. This output combination generally seems to work quite well. Even if the driver does not have much time for studying the display, many drivers still seem to appreciate the availability of display output. The advantage is that the (static) text and graphics on the screen remains there long enough for the driver to inspect them at will, which is not the case with speech.

Figure 1. Inputting a destination with one of today's car navigation systems.

What is new in VICO as regards navigation is the spoken negotiation of where to go. For input, today's navigation systems require a remote control which the driver uses to specify the destination through prolonged interaction with the display, doing spelling, on-screen navigation, between-screens navigation, etc., cf. Figure 1. This is definitely not very traffic-safe to do. Spoken interaction will change that, of course, but, very likely, the spoken output during destination negotiation could benefit from being supported by output on the display as long as interaction mainly takes place through speech.

We decided to investigate if and when the driver might want to look at the display during a destination negotiation dialogue with VICO, as well as the kind of information users might want on the display. We made a series of Wizard-of-Oz (WOZ) experiments with spoken input and spoken and text output in December 2001 (3 subjects) and January 2002 (10 subjects), see [Bernsen and Dybkjær 2001] for details on the December WOZ experiments. To simulate driving the car in traffic we used a PC car game. Subjects were seated in front of a 42" flat screen display showing the traffic ahead in wind-screen view and the traffic behind in rear-mirror view, cf. Figure 2. To control the car, subjects had a force-feedback steering wheel and pedals (accelerator and brakes). Next to the large screen was a small portable computer simulating the in-car display and showing system output text. For the spoken output, the Festival synthesiser was used. Each user was asked to carry out three scenarios. Subjects were interviewed after their interaction with the system.

The dialogue with VICO was in English. The scenarios all concerned route planning for Danish destinations. We experimented with three different text versions on the car display. All three versions were displayed to each user (one version per scenario) but in differing order. One version was a full text repetition of the spoken output, a second version only included the key destination items of the spoken output, and the third version only provided the agreed-upon destination as text output at the end of the dialogue.

The experiments aimed to collect data on which among the kinds of information offered users would like to see on the display and in which situations they would look at the display. In the following we describe the findings.

Many subjects found it less stressful to use the car game than to drive a real car. As a major reason they indicated the fact that they knew that nothing would happen even if they crashed the car. However, some subjects found that it required much more concentration to play the car game than to drive a real car. Typically, these subjects also found it unsafe to look at the car display whilst driving.

Figure 2. Driving the simulated car.

Although most subjects found it less stressful to play the car game than to drive a real car, only a couple of them stated that they used the display quite frequently. Most subjects did not use it much, either because they found it unsafe or because they did not feel a need for it. When the display was used it was typically to cross-check the spoken output. The quality of portions of the output speech was fairly low. Danish location names pronounced by an English synthesiser are often rather difficult to understand. Moreover, as an important part of the experimental setup, the drivers were from time to time distracted by a passenger talking to them, which meant that they were likely to miss what was being said by the system. Better synthesis is certainly available which will reduce the first problem whereas the second problem is not likely to go away in real driving situations.

When subjects did not hear what the system said or were not sure that they got it right, they would typically either ask for repetition of the spoken output or look at the display. Since most subjects only used the display infrequently, these users were not aware of the changing amount of feedback provided in the different experimental conditions. A couple of users complained that there was nothing on the display when they looked. They probably checked the display when the system provided the shortest version of its text output, i.e. when only the finally agreed destination would be displayed but nothing would be displayed during the negotiation dialogue.

Although, for the reasons stated, we did not collect that much data on user preferences as to the length and contents of the text displayed, it appears from the subsequent discussions with subjects that the medium-length text version was the most appropriate. The short version does not provide sufficient support since there will not always be information available on the display when the driver has missed the spoken output. The long text output version, on the other hand, includes too much superfluous information and may be hard to interpret at a glance. The long version may therefore be deemed less safe and less to-the-point than the medium-length version which contains the key destination information expressed in as condensed a form as possible. The main functions of the text output are to allow the driver to check correctness of understanding of the spoken output and to check up on the present state of progress of the dialogue. Even if oral dialogue requires less effort from the user while driving than reading text on a display, situations may occur in which the driver stops listening to the system in order to handle a difficult traffic situation or because a passenger speaks to the driver during the spoken system output. Returning to the dialogue, the driver may either catch up by asking the system where they were at or by checking the display. Based on the WoZ results, we expect that there will be individual differences as to which of the two options drivers prefer because our subjects behaved quite differently with respect to their use of the display versus spoken dialogue. People have different driving experience and different habits, both of which are likely to influence their preferences, so both options should probably be enabled.

Inherent to the problem discussed in the present section is another, larger and more troublesome issue which we have not even begun to analyse. Whatever speech-in-the-car purists might argue, the in-car display is perhaps not likely to go away unless prohibited by law. It is simply too useful for presenting all kinds of information to the driver. If this is true, it becomes less clear exactly how much we will have achieved with respect to increasing traffic safety by replacing destination entry by remote control by destination negotiation through spoken dialogue. To be sure, something will have been achieved since destination entry by remote control is lengthy and hazardous but how much depends on the driver's additional use of the display.

5. WHO IS DRIVING THIS TIME?

Based on its observations on the driver's behaviour, VICO incrementally builds and uses for on-line adaptation to the driver a user model for each driver of the particular car in which the system is installed (see Section 6). In

order for VICO user modelling to be of any use, VICO must be able to determine which of the car's drivers is currently driving. Furthermore, driver identification has to be made with *near-certainty*. If it is uncertain that VICO has correctly identified the driver, driver misidentification will happen too often. In such cases, the driver is likely to be "mistreated" because VICO will adapt its dialogue behaviour to the driver based on a wrong user model. Similarly, the modelled behaviour of the misidentified driver will tend to fudge up the misallocated user model with misleading information. In addition, since the driver's user model cannot be invoked until the driver has been identified, VICO must identify the driver either as one already known to the system or as a new driver *up front*, i.e. before or, at the latest, as soon as that driver starts the dialogue. Later identification means less support for the driver, and the updated user model runs the risk of having missed to collect important information on the driver's behaviour.

In SLDSs, driver identification design is a non-trivial problem. Among the many conceivable options, we have considered the following [Bernsen 2002]:

- *voice identification.* Even though today's voice identification technology is not perfect, it might be possible to get near-certain identification in the car, simply because most cars, with the exception of rented cars, have rather few drivers. Voice identification is also to some extent an elegant solution because the driver does not have to do anything other than speak to VICO about some task. Voice identification happens as soon as the driver speaks. However, given its less than 100% reliability, voice identification requires the system to provide feedback, so that the user can make sure that correct identification was made. To provide feedback, the system must be taught, once and for all, to associate some output expression, i.e. a code or the driver's name, with the driver's speech signal;
- *driver's code.* Contrary to voice identification, the driver's code must be input to VICO explicitly. This may be done by voice, haptically through keystrokes, through personalised ignition key identification, through biometrics, such as measuring the driver's weight, etc. As we are not assuming the presence of keyboard and biometrics facilities, simplicity and traffic safety speak for acoustic or ignition key codes per driver. Correct driver identification is guaranteed, in principle, in the ignition key case which also does not require the driver to remember yet another code. Given its less than 100% reliability, voice code entry, like voice identification, demands that VICO provides code entry feedback;
- *driver's spelled first name.* This solution enables non-coded feedback on driver identification. However, spelling part or whole of one's name before each interaction is an awkward thing to do;

- *voice name enrollment.* A first-time user speaks his/her name a few times, causing a voice model of the name to be generated. During later use, the driver just has to speak his/her name to get identified. However, since the enrolled name is for recognition purposes only, it cannot be used by the system for providing verification feedback to the driver.

The discussion above aptly illustrates how a range of solutions can be so densely packed in design space that it becomes hard to determine which solution is the best one. However, it may be concluded that the problem of driver identification actually can be useably solved.

An important issue which has not been discussed above, is that *passengers* might want to talk to VICO as well, for instance in order to relieve the driver of having to carry out a lengthy hotel reservation task. Normally, a car has fewer different drivers than different passengers. If the latter also talk to VICO, the system might come to include dozens of user models for a single car, most of which are not being used at all since they were created by one-time passengers. For this reason, an additional requirement on driver identification would seem to be that only drivers, and not passengers, should cause the creation and use of user models when speaking to the system. Several of the solutions discussed above are compatible with this requirement.

6. MODELLING THE DRIVER

Once the driver starts speaking to VICO, the system must try to identify the driver and retrieve the driver's user model, if any. If identification fails, VICO assumes that the driver is new to VICO and immediately creates a new user model (UM) for that driver. In both cases, VICO will subsequently collect relevant information on the driver's behaviour during the spoken interaction and use that information to update its model of the driver. The model itself is used to support the driver during dialogue with VICO. Slightly more systematically expressed, VICO's *generic* UM-related tasks are [Bernsen 2002]:

1. identify the present driver (cf. Section 5);
2. retrieve the present driver's user model;
3. optionally: create a new user model UM(Dx) for a new driver, Dx;
4. make appropriate on-line use of the present driver's user model during the driver's dialogue with VICO;
5. collect new information on the present driver during the driver's dialogue with VICO;
6. update the present driver's user model with the new information gathered;
7. store the user model whenever it has been updated with new information.

From a design viewpoint, and ignoring the issue of driver identification discussed above, the hardest problems in the list above probably are points 4, 5, and 6. The comparative difficulty of points 4, 5, and 6 depends on the nature of the information on the driver dealt with by the UM. Thus, before addressing those problems in more detail, decision must be made on *which type(s) of information* on the driver the system should collect, store, update, and use. In fact, this problem appears to be the hardest of all.

The reason is a rather general one. When embarking on adaptive user modelling in VICO, we enter a technical area fraud with difficulty and past failure. Observation-based user modelling for on-line adaptation appears to be among the most difficult things to do in developing interactive computer systems, independently of whether those systems use speech or other modalities of information representation and exchange. In fact, user adaptation has proved so difficult to do that it seems fair to say that, by and large, and despite numerous attempts in the past 15-20 years, research and industry have had limited success in developing useful adaptive functionality in the huge variety of interactive systems that already exist. There are successful exceptions, of course, but these tend to be functionally simple. The conclusion we should draw from that fact is that we must be extremely careful in selecting the kind of information on drivers which we want to model. It is better to succeed with one, or a few, observation-based adaptive functionalities in VICO than to fail through ignorance of the difficulties involved by trying to develop an unrealistic number of poor adaptive functionalities.

Based on analysis of some 25-30 candidate kinds of information about driver behaviour which VICO might conceivably collect and use on-line, we may distinguish between several different generic types of information which VICO could collect and use adaptively. According to the following, possible typology of information, at least three different generic kinds of information about particular drivers may be distinguished:

1. information on the driver's task objectives due to task goals, preferences, habits, etc.;
2. information on the driver's communication with VICO;
3. information on the driver's experience of various kinds.

In the VICO context, each generic kind of information subsumes several more specific information types, such as the driver's hotel preferences (1), the driver's difficulties in being understood by VICO due to strong accent or dialect (2), or the driver's experience in using VICO itself (3). In other words, the information typology helps generate a structured space of candidates for observation-based adaptive user modelling on-line.

To further constrain the design choices with respect to the user modelling capabilities of VICO, we have identified the following criteria which should

be satisfied by a particular kind of driver information in order for that information to be collected and used by VICO:

1. include at least one user modelling functionality belonging to each type in the typology of generic information about the driver described above;
2. the chosen user modelling functionalities should be top quality in terms of their usefulness to all or most drivers. Some functionality may even be top quality and meet all other criteria in the present list, but if it is only of interest to a small minority of drivers, it remains questionable whether it should be implemented;
3. the chosen user modelling functionalities should provide genuine driver adaptivity without significant drawbacks;
4. the chosen user modelling functionalities should be possible to implement without extreme or unpredictable effort. The reason for including this clause is not only the obvious one that we do not have the time for putting extreme effort into adaptive user modelling. More importantly, it seems easy to conceive of user modelling tasks which cannot be achieved without some kind of research breakthrough, hence the unpredictability clause (see also below);
5. the chosen user modelling functionalities must be based on clearly verifiable information about the driver. In particular, it is not enough that some observable property of the driver's behaviour *might* be due to a problem which system adaptivity could help with. We need to make sure that the property actually *is* due to that problem and might not be caused by some other problem which we do not address.

Space does not allow presentation of the pros and cons which we identified in the analysis of each candidate on the long list of kinds of driver information which could potentially be modelled for adaptive driver support. An example sub-type of Type (1) in the typology above, i.e. *information on the driver's task objectives*, is: store the driver's past hotel preferences, such as number of stars, price, location (city centre, countryside), hotel chain, and possibly other selection constraints as well. Even if not told about them by the driver, VICO could offer to use those constraints as selection criteria when looking for a suitable hotel. It is important, of course, that the driver is able to override those constraints and provide new ones. If not, all we will be doing is to produce yet another failed attempt at creating useful system adaptivity. However, in this case it is easy for the driver to override VICO's suggestions because the driver will be told that the UM has been used for selecting the hotel(s) options offered.

Let us try to evaluate the hotel preferences user modelling functionality using the selection criteria presented above. The functionality is based on clearly verifiable information about the driver (Criterion 5). Thus, the driver's hotel preferences become apparent in the driver's dialogue with

VICO. Moreover, it is possible to write an update algorithm for a driver's hotel selection UM which only produces hotel selection constraints when a clear pattern can be discerned in the driver's hotel preferences. The functionality under consideration does not appear to have any significant drawbacks (Criterion 3). Also, this user modelling functionality can be implemented without extreme effort (Criterion 4). What this means is simply that, if we want to implement user modelling functionality at all, implementation of drivers' hotel selection preferences would appear very much as the standard case. So, the final question is whether information on the driver's observed hotel preferences is top quality in terms of its usefulness to all or most drivers (Criterion 2). This question is a difficult one, because the answer to the question depends on, at least, (i) how many users of VICO will actually need to book hotels, (ii) how many users will want to do so *en route*, and (iii) how many of those users have systematic hotel preferences. We do not know the answer to that question at this point but would clearly need to find out the best we can in order to be able to rank the user modelling option just described among its competitors.

Generally speaking, compared to the problem of identifying a user modelling candidate for information about the driver's task objectives, it would seem considerably harder to build adaptive user modelling with respect to Type (2) *information on the driver's communication with VICO.* An example is a system which adapts its dialogue behaviour to drivers whose strong dialect or accent makes their dialogue contributions difficult for the system to recognise and understand. One issue, of course, is that we might need two significantly different dialogue structures to accommodate both standard drivers and drivers with strong dialect or accent, making a solution relatively costly to implement (Criterion 4). A second problem is that any solution may be at risk as long as we do not have efficient ways of discriminating between different possible causes of recognition problems. Recognition confidence scores, for instance, cannot tell VICO whether the cause of repeated recognition problems is a strong dialect or accent or something entirely different, such as a driver who regularly talks to passengers whilst having a dialogue with VICO. Similarly, the measurable facts that a driver produces many out-of-vocabulary words or makes unusually many error corrections may be due to many different causes (Criterion 5). On the other hand, solutions to Type-(2) user modelling problems, if we could only find them, clearly might provide genuine driver adaptivity, possibly without drawbacks worth mentioning (Criterion 3). And, even if those solutions might not benefit all or most drivers, they might benefit large fractions of exactly those drivers who might otherwise have great difficulty using spoken language dialogue systems (Criterion 2).

Type (3) information, i.e. *information on the driver's experience* includes, at least, one obvious candidate for adaptive user modelling, i.e. the driver's experience with VICO itself. The idea is to offer up-front information on VICO to all new drivers independently of whether a new driver asks for it or not. VICO is a complex system both in terms of the tasks it can solve and the languages which may be used in addressing the system (cf. Section 2). New drivers are therefore quite likely to benefit from an introduction to the system which includes information on which tasks it covers and how to operate the system. Thus, provision of this information would seem to be top quality in terms of its usefulness to all or most drivers (Criterion 2) as well as providing genuine adaptivity without any significant drawbacks (Criterion 3). This assumes, of course, that drivers are identified with near-certainty, as discussed above. Implementation will be relatively simple because the system only needs to determine if the current driver is new to the system. It does not have to store a record of the driver's past behaviour nor does it need UM update algorithms, such as those needed for hotel preferences (Criterion 4). Finally, as argued in Section 5, it is clearly verifiable if the driver is new to VICO or not (Criterion 5).

7. CONCLUSION AND FUTURE WORK

In this paper we have discussed four issues of importance to future in-car information systems development. At the time of writing, only one-and-a-half of these issues have been resolved to our satisfaction. This is true, firstly, of the issue of which driver-system dialogue-relevant information to present on the in-car display.

The half solution found concerns observation-based user modelling for on-line adaptivity. We have designed and implemented the hotel selection preferences UM discussed in Section 6 above, see also [Bernsen 2003]. The functionality still needs to be tested with real users, however.

As for the two other issues discussed above, i.e. driver identification and how and when to make VICO listen, we are still investigating the pros and cons of different solutions. Thus, the duration of the time window in which the system should be listening to the driver will form the topic of future experimentation. Similarly, the problem of driver-passenger conversation while the system is listening continues to demand a more efficient solution than any we have investigated so far. As regards driver identification, we are investigating what the most elegant, useful and usable solution might be. The same applies, in part, at least, to the issue of adaptive driver modelling. Potentially, adaptive driver modelling could be extremely useful to drivers, yet the complexity of the options, trade-offs and technical issues involved

would seem to make adaptive driver modelling a highly interesting research challenge which is likely to occupy researchers for some time until the terrain has been appropriately charted and useful solutions identified.

8. ACKNOWLEDGEMENTS

VICO is supported by the EU HLT (Human Language Technologies) Programme under Contract IST-2000-25426. We gratefully acknowledge the support.

9. REFERENCES

Bernsen, N. O.: Report on User Clusters and Characteristics. VICO Report D10, NISLab, 2002.

Bernsen, N. O.: On-line User Modelling in a Mobile Spoken DialogueSystem. Proceedings of Eurospeech 2003 (to appear).

Bernsen, N. O., Dybkjær, H. and Dybkjær, L.: Designing Interactive Speech Systems. From First Ideas to User Testing. Springer Verlag 1998.

Bernsen, N. O. and Dybkjær, L.: Exploring Natural Interaction in the Car. In Bernsen, N. O. and Stock, O. (Eds.): Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue, Verona, 2001. ITC-Irst, Trento, 2001, 75-79.

Furui, S.: Speech Recognition Technology in Multimodal/Ubiquitous Computing Environment. This book.

Manstetten, D., Berton, A., Krautter, W., Grothkopp, B., Steffens, F. and Geutner, P.: Evaluation Report from Simulated Environment Experiments. VICO Report D7, DaimlerChrysler, 2002.

Minker, W., Haiber, U., Heisterkamp, P. and Scheible, S.: Design Issues and Evaluation of the Seneca Speech-based Human-Machine Interface. Proceedings of the International Conference on Spoken Language Processing (ICSLP'2002), Denver, USA, 2002, 265-268.

Salmen, A.: Multi-Modal Menus and Traffic Interaction. Timing as a Crucial Factor for User Driven Mode Decisions. Proceedings of the Language Resources and Evaluation Conference (LREC'2002), Las Palmas, 2002, 193-199.