Project ref. no.	IST-2000-25426
Project acronym	**VICO**
Project full title	**V**irtual **I**ntelligent **Co**-Driver

Security (distribution level)	Project internal
Contractual date of delivery	May 2002 (M 14)
Actual date of delivery	1.8.2002
Deliverable number	D10
Deliverable name	Report on user clusters and characteristics
Type	Report
Status & version	Draft, 17.7.2002
Number of pages	28
WP contributing to the deliverable	WP 5.4
WP / Task responsible	NISLab, University of Southern Denmark
Other contributors	-
Author(s)	Niels Ole Bernsen
EC Project Officer	Domenico Perrotta
Keywords	user modelling, user-adaptive spoken dialogue systems, in-car information systems
Abstract (for dissemination)	This report from the IST/HLT VICO (Virtual Intelligent Co-driver) project presents an analysis of the use of explicit user behaviour and user property modelling for the design of on-line adaptive spoken dialogue in-car information systems.

Formateret

Table of contents

Report on user clusters and characteristics

Niels Ole Bernsen, NISLab

1. Introduction

1.1 Purpose and plan of this document

This document on user clusters and characteristics provides first documentation on the VICO user modelling module. The purpose of the VICO user modelling module is to collect, store, update, and make available for on-line use information on each user of a particular in-car VICO system. On-line, this information will be used to adapt the dialogue conducted by VICO to the individual drivers using the system. This document presents a rather general analysis of user modelling for in-car information systems, the analysis being restricted primarily by the scope of the, admittedly many and highly diverse, tasks which VICO is planned to, or may, help drivers solve.

It is useful to distinguish between generic and specific user modelling tasks. A *generic* user modelling task is a task which probably should be performed by any user modelling module for in-car information system use, independently of the type of information which is being modelled. A *specific* user modelling task is a user modelling task which addresses a particular type of information on users (drivers).

Below, Section 2 presents the generic user modelling tasks which must be solved by VICO. These tasks are analysed in more detail in Section 5. Section 3 analyses the types of information about drivers which might be collected and used in the specific VICO user modelling tasks and proposes a typology of those information types. Structured according to the typology, Section 4 first proposes a set of criteria for selecting the specific user modelling tasks to be implemented in VICO, and then discusses the pros and cons of a rather extensive series of specific user modelling tasks which VICO might address. Since many open questions remain, Section 4 concludes with a partial list of specific user modelling task candidates. Finally, Section 6 briefly describes a few generic issues concerning the VICO user modelling module.

This document assumes that VICO user modelling will be implemented in two stages. At the first stage, due September 2002, we will implement a single specific user modelling functionality (for a single information type) together with most of the generic functionality needed for user modelling in general. The phrase “most of” is explained in Section 2 below. At the second stage, we will implement the remainder of the general user modelling apparatus as well as several additional specific user modelling functionalities (information types). The present document aims to identify the candidate specific user modelling functionality to be implemented at the first stage and to open VICO consortium-wide discussion of the specific user modelling functionalities to be implemented at the second stage. Thus, the present document is a discussion document and comments and suggestions are cordially invited from everybody in the VICO consortium.

Based on the present document, a high-level user modelling module specification has been developed. Refinement of the specification has resulted in a program specification which will enable implementation of a first user modelling module version for the first VICO prototype due in the autumn of 2002.

1.2 On user modelling for adaptive system behaviour

In addressing the VICO user modelling issue, we enter the treacherous terrain of user adaptation and customisation. The difference is that *customisation* is done off-line whereas *adaptation* is done on-line. Thus, to customise VICO, the driver might, for instance, make certain adjustments to VICO prior to interacting with the system, such as selecting the language of interaction or selecting a female or male output voice. On the other hand, it is VICO itself which, based on its own observations of the driver, adapts to that driver. In general, customisation is the less risky development endeavour compared to adaptation. In customisation, we develop what we consider to be helpful options for the user to select from. If we do this in ways which are less helpful to users than they perhaps might desire, users are sometimes able to de-select some or all of the customisation options offered by the system. In adaptation, however, the user normally has no choice but to be exposed to the system's adaptation strategies. If these are not helpful, or are even counter-productive, all the user can do is to develop a more of less profound dislike of the system - unless, of course, one adds the customisation option of switching off the system's user adaptation functionality.

For the sake of clarity, it may be useful to point out that the term "adaptation" – and, correspondingly, the terms "user model" and "user modelling" - is being used in many different ways in the spoken dialogue systems literature. In this document, we are only addressing *explicit user model-based adaptation,* i.e. adaptation to users based on separate (implemented and hence explicit) models of each individual user, the models being constructed from observation of the users and their behaviour during dialogue with VICO. The phrase "users and their behaviour" has been chosen deliberately to encompass the user modelling options considered below. That is, some of these user modelling options are certainly based on user observation but it is questionable how much observed user *behaviour,* if any, is needed for the model to do its work. However, the VICO system will do much adaptation which is not based on explicit (implemented) models of individual users. For instance, the speech recognisers used in VICO are able to adapt on-line to different properties of the spoken input, such as low-pitched voice and high-pitched voice; the task-dependent VICO dialogue structures have been designed for graceful degradation on-line when facing problems during communication with the users; lots of design time (off-line) adaptation to user behaviour has gone into the design of VICO's dialogue structures; etc. It is a common characteristic of all these forms of user adaptation that they are not based on explicit models of VICO's individual users, which is what distinguishes them from the main topic of this document.

Customisation is mostly ignored in the following except in cases in which customisation would seem to be preferable to adaptation. Technically, customisation would seem rather close to product implementation, and we know little about which form this will, or may, take. For instance, customisation of VICO to a specific language or output voice assumes that the product version of VICO will enable such things to be done and will provide interactive ways of doing them.

Adaptation is another matter, however, which is addressed in what follows. At the same time, on-line system adaptation to users is among the most difficult things to do in developing interactive computer systems, independently of whether those systems use speech or other input/output modalities and independently of their domain of application. In fact, system adaptation to users has proved so difficult to do that it seems fair to claim that, by and large, and despite numerous attempts in the past 20 years, at least, research and industry have failed to develop useful adaptive functionality in the huge numbers of interactive systems which already exist. There are successful exceptions, of course, but these are few and tend to be functionally simple. The conclusion we should draw from this fact, it seems, is that we must be extremely careful in selecting the specific user modelling functionalities we want to build. It is probably better to succeed with one, or a few, adaptive functionalities in VICO than to fail through ignorance of the difficulties involved by trying to develop an unrealistic number of poor adaptive functions. Readers of this document will perhaps tend to agree that the area of adaptive VICO user modelling is replete with pitfalls many of which are not apparent at first sight.

2. Generic VICO user modelling tasks

To enable user model-based adaptation, VICO must implement the following tasks, at least:

1. identify the present driver;
2. retrieve the user model of the present driver;
3. create a new user model (UM_Dx) for a new driver, Dx;
4. make appropriate use of the driver's user model during dialogue with VICO;
5. collect new information on the present driver from the driver's dialogue with VICO;
6. update the present driver's user model with he new information gathered;
7. store the updated user model.

These tasks, it seems, must be handled independently of which aspect(s) of (or types of information on) the driver the user model addresses. Each task can be designed and implemented in different ways. Some tasks are relatively straightforward to deal with, such as (2), (3) and (7). Others are more or less complex, depending on the type of information addressed, notably (4) and (6). Also (1) and (5) are liable to demand some difficult design decisions.

In the first VICO prototype, we will implement the generic apparatus for (3), (4), (5) and (6) above as well as a, possibly temporary, version of (2) and (7). Driver identification (1) needs more investigation before we can decide on the best feasible approach to adopt (see Section 5.1). User model retrieval (2) and storage (7) *may* not be implemented in its final form at this stage. What this means is that, in the first VICO prototype, user models will be stored in the dialogue manager which is already involved in doing all other processing of the user models, the user modelling module being a module within the dialogue manager. Thus, in the first prototype, user model retrieval and storage is not an issue. For the second VICO prototype, however, it is possible that user models will be stored on the Car Wide Web (CWW) and not in the dialogue manager, and that individual user models will have to be retrieved from the CWW before the dialogue manager can use them. If decision is made to do things this way, for instance because information storage turns out not to be appropriate for the dialogue manager to do, this will require some additional but straightforward implementation work.

3. VICO user modelling information types

Central to VICO user modelling are the kinds of information on the driver which the system collects and makes use of. Depending on the kind(s) of information addressed, user modelling and the resulting system adaptation to users may be easy or difficult to do, highly useful to the intended users or more or less detrimental to their perception of the system. In other words, it cannot be taken for granted that, given some type of information on drivers, Ix(D), there is always a user modelling solution which will be beneficial to the intended users. This is due to the fact that the system's capabilities of user observation are still rather limited. Clearly, VICO does not yet have the capabilities of observation that humans have. For instance, VICO can relatively easily be made to observe the amount of misrecognitions which are made of a particular driver's spoken input, or the amount of system- and/or user-initiated error correction meta-communication which occurs during that driver's dialogue with the system. However, misrecognitions and error correction meta-communication may be the product of many different and highly diverse causes, such as a speech disorder, a strong dialect or accent, poor mastery of a particular language more generally, or driver cross-talk with passengers during dialogue with VICO. As long as VICO does not possess the capability of correctly diagnosing those causes, any attempt at adequately adapting the system's behaviour to the driver may fail. In addition, the degree of complexity of specification and implementation is no sign at all of the eventual usefulness to the driver. At this early stage in the development of user model-based adaptive spoken dialogue systems, the most successful solutions might well turn out to be the simplest ones.

3.1 A first list of information types

Let us begin by looking at the user modelling information ideas in the VICO Work Programme. These ideas are described in the following list of potential user properties to be modelled by VICO, systematising them a bit:

- habits, e.g. driving style, types of streets used;
- preferences for, e.g., hotels, restaurants, type(s) of tourist information, news topics;
- experience, e.g. with the VICO system itself (expert, casual user, novice);
- level of expertise in, e.g., the particular foreign languages known to VICO, usage of in-car devices and services (apart from VICO itself, cf. above);
- characteristics, e.g. native language, age, voice properties;
- geographical profile, i.e. information about previously or frequently requested route descriptions and locations, home;
- attitude towards VICO, e.g. talkativeness, scepticism, politeness;
- current situation, e.g. stress, emotion, in a hurry or at ease, economical or not.

This is a long and interesting list even if it does not exhaust the potential for VICO user modelling. Even if we choose to remove some of the options above because, on reflection, these may not be evident candidates for user modelling, the list will remain a rather long one. At this point, and inviting corrective information from the VICO consortium, less obvious candidates for user modelling include:

(a) driving style. It is not clear at this point how to define, measure, and use information on the driver's driving style;

(b) types of streets used. So far, we have no idea of how to obtain information on the types of streets used by a driver nor what to do with that information if it could be obtained. For instance, does a driver who normally prefers minor streets to larger ones also prefer those minor streets when in a hurry? If not, how would VICO ascertain that the driver is in a hurry?

(c) level of expertise in usage of in-car devices and services. For the second VICO prototype it is planned to incorporate car manual functionality. This functionality might alleviate the need to assess particular drivers' level of expertise in using in-car devices and services;

(d) age. It is not clear how to measure this property of drivers which, rather, would seem to be a customisation property. Still, once told, or having discovered, the driver's age, what might VICO do differently?

(e) geographical profile, i.e. information about previously or frequently requested route descriptions and locations, home. This one is not entirely clear. In particular, what is meant by "route descriptions"? However, given the ease with which we expect drivers to tell VICO where to go, it is not entirely obvious that it is worthwhile implementing a functionality by which drivers could shortcut the entering of destinations. This functionality, it seems, is far more useful in today's navigation systems where drivers have to enter destinations through spelling using a remote controller. Still, let us keep this information type for further discussion below;

(f) current situation, e.g. stress, emotion, in a hurry or at ease, economical or not. It seems rather clear that drivers who are in a hurry may prefer the fastest route to their destination. Similarly, economical drivers (who, in addition, are not in a hurry) may generally prefer the most direct route whether or not this route is the faster for reaching the destination. By nature, however, current situations change. It is therefore not at all evident that current situations are entities which can be predicted by a user model which has been built from past observations of the driver.

It is useful at this point to insert a *disclaimer,* however, especially wrt. (f) above but also wrt. some of the user modelling options discussed below. It is that not all useful user modelling has to be based on historical records of past interactions with VICO. Some user modelling functions may be of the following nature: watch the driver's interactions with VICO (only) *until* it is clear what the problem is and then take immediate action. In such cases, historical records of past interactions in the user modelling module are not necessarily relevant. Rather, the dialogue history from the ongoing dialogue must be used. Again, the problem here is likely to be that VICO will possess limited diagnostic capabilities which could enable detection of the problem in-session with VICO. To the extent that this problem can be solved, new possibilities open up for VICO user modelling based on the (current) dialogue history rather than on records of past interactions. This topic merits careful investigation in the second phase of the VICO project.

3.2 Revised list of information types for user modelling

In this section, the list presented in Section 3.1 has been reduced taking into account the comments made above. It should be noted, however, that the VICO consortium may have arguments for re-inserting removed items into the list below as well as new ideas for types of information which are promising candidates for user modelling in support of adaptive system behaviour. Several types have been added to the list below, including point of interest information, strong accent or dialect, speech disorder, speaking style, indecisiveness, and toleration of external disturbance. Some of these have several sub-categories which will be discussed below.

- preferences for, e.g., hotels, restaurants, tourist and point of interest information, news topics;
- experience, e.g. with the VICO system itself (expert, casual user, novice);
- level of expertise in, e.g., the particular foreign languages known to VICO;

- characteristics, e.g. native language, voice properties, strong accent or dialect, speech disorder;
- attitude towards VICO, e.g. talkativeness, scepticism, indecisiveness, politeness, speaking style, toleration of external disturbance;
- geographical profile, i.e. information about previously or frequently requested route descriptions and locations, home.

A final conclusion which may be derived from Section 3.1 is that, when looking for candidates for VICO user modelling, we are faced with a very large search space. It is important that we search this space carefully. The best candidates (driver characteristics) for modelling may not yet have found their way into the present document.

3.3 Generic information types

The VICO user modelling module should collect and store information about each driver, which is highly relevant and significant to facilitating that driver's goals and dialogue with VICO. The list in Section 3.2 is not a very principled one. It is useful to have a general typology of the kinds of information VICO user modelling might address. Subject to the qualifications stated in Section 3.2, the following typology may be useful:

1. information on the driver's task objectives: preferences for, e.g., hotels, restaurants, tourist and point of interest information, news topics, address locations;
2. information on the driver's communication with VICO: native language, mastery of foreign (VICO) languages, voice properties, strong accent or dialect, speech disorder, attitudes such as talkativeness, scepticism, indecisiveness, politeness, speaking style, toleration of external disturbance;
3. information on the driver's experience with VICO itself (expert, casual user, novice).

Given the questions raised in Section 3.1, it is clear that this typology may not (yet) be exhaustive. The typology needs to be tested with any reasonable information type example we can think of. However, the typology does seem to capture the examples provided in Section 3.2. As the typology may not be immediately obvious, some explanations follow.

(1) Information on the driver's task objectives: this is information about what the driver wants to do as shown in the dialogue history of ongoing dialogues with VICO. So, for instance, a particular driver only uses VICO for route tasks, another driver uses VICO a lot for car device information, or some driver always wants three-star hotels. Note that (1) has several levels of generality: the driver prefers a single task and does that a lot, the driver prefers a few particular tasks, or, within a particular task, the driver prefers particular choices.

(2) Information on the driver's communication with VICO: this is information about how the driver communicates with VICO independently of the driver's task objectives. So, (2) is orthogonal to (1). (1) and (2) are similar, however, in that they are both based on observations recorded in the dialogue history.

(3) Information on the driver's experience with VICO itself. The reason why this entry does not seem reducible to (1) is that experience with VICO, including knowledge of what VICO can do and how to operate VICO, is not obviously something which can be deduced from the dialogue history. For instance, if the driver only uses VICO for the route task, this may be because the driver does not want to make use of the other functionality which VICO makes available, but it

may also be because the driver does not know that VICO can do things other than helping with the route task. However, VICO will at least know if the driver is a new driver for which VICO has just created a user model. So, it is actually possible for VICO to ascertain that the driver *may* not have much experience with systems such as VICO.

One reason why an exhaustive typology of generic information types could be useful, is that we might use it to help decide which user modelling functionalities to offer. For instance, it might be a good strategy to aim to demonstrate at least one user modelling functionality per type. In this way, we might claim that we have selected the user modelling functionalities we have in order to demonstrate how to do user modelling concerning each one of the types in the typology, rather than randomly choosing the user modelling functionalities which we want to demonstrate.

4. How to select user modelling functionalities

This section discusses rules or principles for use in deciding which user modelling functionalities to demonstrate in VICO. Intuitively, perhaps we should not select a user modelling functionality just because it can be done. Even if it can be done, it may not be technically feasible to construct the functionality in a form which is clearly useful to drivers, or, even if this is possible, the proportion of drivers who may benefit from it may be likely to be quite small. Thus, summarising the argument so far, and perhaps making it a bit more precise, the proposal is that we select the user modelling functionalities to be designed and implemented according to the following criteria:

1. we would like to include at least one user modelling functionality belonging to each type in an exhaustive typology of generic types of information about the driver;
2. the chosen user modelling functionalities should be top quality in terms of their usefulness to all or most drivers;
3. the user modelling functionalities should provide genuine driver adaptivity (or observation-based user assistance) without significant drawbacks;
4. the user modelling functionalities should be technically feasible and possible to implement without extreme effort (since we do not have the time for putting extreme effort into them);
5. the user modelling functionalities must be based on clearly verifiable information about the driver, for instance information which can be gathered from the dialogue history.

The drawbacks mentioned in (3) above are important. The reason why adaptivity has mostly failed in the past is that historical information about user behaviour tends to ignore that user behaviour changes over time. Like somebody wrote (I believe it was the German philosopher Edmund Husserl): “Historical evidence is excellent evidence, but only for the making of historical conclusions”. Thus, if the system changes its interaction based on how the user has behaved in the past, the user very often finds that the change in interaction made by the system is a change for the worse because it is in conflict with what the user wants to do right now. We need to try to avoid flawed VICO interactivity of this kind. Drawback analysis, therefore, is an important part of the game when designing system adaptivity.

Another point worth highlighting is the one made in (5) above that we need clearly verifiable information about the driver. The following discussion will show that this criterion is sometimes hard to meet. In such cases, we face the choice of improving on the state-of-the-art in order to be able to meet the criterion, or drop the user modelling functionality in question.

Let us now look more closely at a range of candidate user modelling functionalities and how these measure up to the criteria above. The discussion will be necessarily brief in cases where internal VICO consortium discussion has not yet provided the information required for proper assessment.

4.1 Information on the driver's task objectives

Information on the driver's task objectives could facilitate the driver's dialogue with VICO in various ways. However, some of these possible ways come with drawbacks which are highlighted below.

4.1.1 Hotel information

The VICO user model could store the driver's past hotel preferences, such as hotel name and location, number of stars as well as the other selection constraints used in the hotel reservation task. Even if not told to do so by the driver, VICO could use those constraints as selection criteria when finding a hotel for the driver. It is essential, of course, that the driver is able to override those constraints and provide new ones. This is possible to do because the driver will be asked when VICO proposes any hotels meeting those selection constraints. The only potential **drawback** apparent at this point may be that only a minority of the drivers using VICO are relatively frequent users of VICO's hotel reservation functionality. A problem is that we do not quite know how to find out to which extent this is true. Thus, **4.1.1 might meet our user modelling selection criteria.**

4.1.2 Restaurant information

Due to unplanned loss of time on other VICO tasks, restaurant reservation may not be implemented. It is therefore uncertain that VICO will be able to collect information on drivers' restaurant preferences. Structurally, the restaurant reservation task is similar to the hotel reservation task and might therefore be implemented in the VICO product development phase following the approach demonstrated for the hotel reservation task.

4.1.3 Tourist and point of interest information

This type of information offers several user modelling options. One is to store information on the *scenic routes* which drivers appear to prefer. This task has not yet been addressed in the project but, possibly, it will be structurally similar to the point of interest (POI) task which is being specified at the time of writing. So, we will discuss the POI task in the following.

The VICO user model could store the driver's preferences for *certain types of point of interest*, i.e. types of POI which it makes sense to want to know more about out of, e.g., historical interest, such as castles and churches. Thus, if the driver has in the past asked to hear about the churches which the car passes by, VICO could offer to tell about churches by default when the driver wants to have en-route location-dependent information. Like hotel preference (4.1.1), POI information preference user modelling is not applied by going behind the back of the driver but involves asking the driver for confirmation before acting on the user-adaptive information. **4.1.3 might meet our user modelling selection criteria.** In fact, there are strong similarities between POI information preference user modelling and hotel preference user modelling (4.1.1), as well as between POI information preference user modelling and the modelling of user preferences with respect to scenic route items.

As another example which might as well have been included in Section 4.1.5 below on address locations, VICO might store the driver's *petrol station brand preference,* petrol stations being an example of a POI type. For instance, if in the past the driver has used only the brand name Shell when providing VICO with a petrol station brand name, then VICO could look for Shell stations and offer the nearest one whenever the driver wants to go to a petrol station, possibly also telling the driver about the petrol station which is nearest the driver if that station happens not to be a Shell station. People tend to stick to a particular brand of petrol for some length of time if they have this preference in the first place, often because of price reduction arrangements with a particular petrol station chain. And if they change brand preference, they tend to stick to the new one for some time. A possible **drawback** in choosing user modelling of petrol station brand preferences is that these perhaps are not top quality in terms of usefulness to all or most drivers. It may happen quite often that the driver prefers the nearest petrol station independently of its brand, simply because the driver needs a filled tank right away. How often this is the case we do not know and we do not quite know how to find this out either. A possible **second drawback** is that the databases used in VICO do not appear to be entirely consistent in the way they deal with petrol station brand names. Thus, even if all drivers need petrol, only some of them adhere to a particular brand of petrol and only if their needs are less than urgent. Furthermore, petrol station navigation is a relatively simple task. Comparing this with hotel reservation, fewer drivers are perhaps likely to need to make hotel reservations with VICO but the hotel reservation task is a far more complex one. These are the trade-offs we have to deal with when selecting the specific user modelling tasks to be implemented in VICO, often doing so in the face of incomplete information about the actual behaviour of the drivers we intend to support in the best way possible.

4.1.4 News topics

The driver's preferences for particular news topics or news media are evident candidates for user modelling. So, **4.1.4 *might* meet our user modelling selection criteria.** However, it is too early at to discuss news preferences modelling as we have at present little specific information on the news reading task which will form the basis for user modelling of drivers' news reading preferences.

4.1.5 Address locations

Modelling of drivers' address preferences is a tempting candidate for VICO user modelling. The point in doing this is not so much, perhaps, to save drivers the relatively little effort it is to tell VICO where to go. Rather, knowledge about the *generic address locations* (regions, communes, large cities, etc.) the driver has driven to, or usually drives to, could be used, for instance, to upload vocabularies and lexicons for a particular area if the driver normally drives in, or into, that area. This could facilitate initial recognition and natural language processing (NLP) of the driver's navigation goals. A potential **drawback** is that if a significant number of drivers break the pattern relatively often, this kind of adaptivity might create as many problems as it solves. A problem is that it may be difficult to find out about the seriousness of that potential drawback. In addition, deeper analysis shows that there is a **second drawback**. It is that address preferences modelling is a higher-risk user modelling candidate than many others because it does not seem to make sense to *ask* the driver before uploading particular vocabularies and lexicons. Some other location candidates discussed above assume that the driver is being asked, explicitly or implicitly, about something before VICO takes action, such as "Do you want a three-star hotel?", or "There are two 3-star hotels in Corvara. Do you want one of these?" This difference seems to be due to at least two factors: (i) VICO's address items vocabulary is too large to be simultaneously active, and (ii) whereas drivers are likely to go directly into the address (or route) task by providing VICO with some or all of the address items they have in mind, tasks such as the hotel reservation task are both more special and more complex than route navigation, which is why we may expect them to require several exchanges between the driver and VICO. This, again, makes it possible for VICO to make use of its stored knowledge of user preferences after the user's first turn but before the driver has exhausted the input needed to complete the task.

A second possible use of the driver's observed address location preferences is to store the *specific address locations* to which the driver has driven in the past. For instance, *if* the driver has been helped by VICO in the past to go to a particular street and street number in Trento, *if* that location is the only one in Trento which the driver has been helped by VICO to go to in the past, and *if* the driver now wants to go to Trento, *then* VICO could ask if the driver wants to go to that particular street number. The same applies to street: when given city and street, VICO could offer the street number if that number is the only one the driver has used VICO to get to in the past. Such a *previous goals list* could facilitate dialogue with VICO. Still, clearly, this is a lot of *ifs* to keep track of. Another possible **drawback** is that, for reasons of privacy protection, a previous goals list might require an editing function, so that drivers could delete previous goals from the list, for instance because they do not want other users of the car to know where they have been. Note, however, that the need for an editing function depends on the way drivers are being identified by VICO. If drivers will be identified through voice identification, for instance, an editing function will not be necessary, whereas if drivers are being identified through codes or passwords, an editing function may be required (see Section 5.1). A **second drawback** is that VICO should be so simple to provide addresses to that a specific address location preference user modelling functionality might not be extremely useful to drivers.

There are probably several other types of user preference information which we could be considered for VICO implementation as well. So, we may be confident to be able to choose at least one of these and thus provide useful adaptation based on the driver's historical task objective preferences.

4.2 Information on the driver's communication with VICO

Information on the driver's communication with VICO is a large mixed bag of user modelling possibilities and opportunities, some of which may be beyond current technical capabilities (cf. Point (5) in the introduction to Section 4). Below, these opportunities are distinguished into language skills, i.e. the driver's native language and mastery of foreign languages; voice, speech, and language problems, such as accent, dialect, speech disorders, etc.; and communication attitudes, such as talkativeness, scepticism, politeness, etc.

4.2.1 Language skills

Potentially, it might be useful to be able to identify the driver's *native language* in the limited sense of finding out if the driver has a native language which is among the languages mastered by VICO. Language identification in general is clearly beyond VICO, not only because general language identification is a difficult research problem in itself but also because making use of this information would require a host of functionality which VICO is not planned to have. However, finding out if VICO does have the native language of the driver could be useful because VICO might then be able to immediately switch to that language. This limited language identification problem has not been discussed in the VICO consortium and it is not clear, at the time of writing, whether VICO will come to have this capability. It may be added that the driver's native language is a clear candidate for customisation rather than user model-based adaptation. The driver might simply be offered the choice between the languages mastered by VICO and then select the appropriate one. Henceforth, VICO would speak to the driver in that driver's native language. This customisation choice might be offered to every new driver. Technically, making the language choice one of customisation rather than of adaptation would make VICO's task far simpler and its solution more reliable as well. Since VICO will know if a driver is new to the system, it would be relatively straightforward to offer the language customisation option to new drivers.

A variation on the language choice problem is the issue of VICO's observing, and acting upon, the driver's *mastery of one or more of the languages used by VICO.* The customisation approach recommended above with respect to the driver's native language might work in this case as well, at least for a start. That is, instead of working hard on making explicit the driver's mastery of various foreign languages, VICO might simply ask new drivers about their preferred language (among VICO's own languages) for communicating with VICO, and henceforth use that language in communicating with that driver. It is clearly possible, however, that some drivers do not adequately master any of VICO's languages but just customise VICO for the language they are best at among those available. In such cases, drivers' lack of language mastery in communicating with VICO may turn out to generate communication problems. These are discussed in Section 4.2.2.

Another possibility, about which little is known at the moment, is that the spotting of problems due to language mastery might, in a limited sense, be mitigated by VICO's use of pronounciation variants for names. However, an approach along these lines might, first of all, require the existence of separate speech recogniser units which include those variants. At the time of writing, we do not know if there will be such units or whether their existence will benefit some drivers in the general perspective necessary for implementing user modelling functionality of this kind.

4.2.2 Voice, speech, and language problems

Potentially as well as in the form of a basic design requirement, VICO has a huge and extremely heterogeneous user population. The only condition that must be met for a driver to get to talk to VICO is access to a car in which VICO has been installed. The implication is that some drivers may have problems communicating with VICO due to voice, speech, and language problems. Apart from the native language identification issue discussed in Section 4.2.1 and other issues to be discovered, a sensible approach is to address all types of information on drivers' communication with VICO on the basis of observed *communication problems or difficulties.* In other words, as long as no significant communication problems are observed in a driver's communication with VICO, nothing is being stored in the user model of that driver. VICO, it would appear, has no reason to analyse, dissect, and store information on drivers' successful communication. All the system has to do about successful communicators is to help them achieve their goals. The implication is that if the driver has no major difficulties in communicating with VICO, the user model for that driver will be empty as regards this generic type of information.

However, as soon as communication problems occur in significant numbers, it is desirable in principle that VICO is able to diagnose their causes and do something about them. An important point to keep in mind when embracing this approach is that VICO's capabilities of diagnosing the causes of communication problems are limited. VICO can be made able to count sub-optimal recognition events, count repair meta-communication events, such as when VICO is not certain what the driver just said and has to produce some turn modification (compared to the turn which would have been produced had VICO been reasonably certain what the driver just said) to make sure, or when the driver initiates repair meta-communication to make sure that VICO's most recent output was correctly understood. The problem, however, is for VICO to correctly diagnose the causes of what went wrong. Low speech recogniser and NLU confidence scores may have many different causes, such as out-of-vocabulary words, too long input sentences, external noise, strong accent or dialect, speech disorders, a driver's cross-talk with passengers, etc. And driver's repair meta-communication may be due to lacking language skills, traffic-induced inattentiveness to VICO's output, etc.

Several kinds of communication problems may be considered, including, at least, recognition problems and NLU processing problems due to:

- strong dialect
- strong accent
- speech problems (disorders)
- very high-pitched voice, very low-pitched voice

One of these problem types, i.e. the high-/low-pitched voice issue, can be dealt with by the speech recogniser. The VICO recogniser will be able to normalise with respect to high and low voice quality using adaptation methods. This means that we can disregard high/low voice quality in the following user modelling analysis.

Based on the dialogue history, the user model could count those difficulties, for instance by counting the number of lower-than-3 recognition scores and/or NLU scores and/or the number of corrections made by the driver. If these are relatively frequent, VICO could alter its dialogue strategy to suit that particular driver, for instance by changing the dialogue structure so as to ask more yes-no questions, ask for one information item at a time, or using other means. The mentioned recognition problems constitute very interesting candidates for VICO user modelling functionality. It seems clear that, in principle, user modelling could help certain drivers a lot. Moreover, the driver characteristics involved do not go away easily. This means that adaptation does not run the risk of the driver suddenly changing behaviour. On the other hand, the kinds of user adaptation needed would seem to be a more serious challenge than the kinds discussed in Section 4.1, because we may need to find ways to change the dialogue structure on-line. This means that we may have to develop several alternative or semi-alternative dialogue structures in order to be able to adapt VICO to different driver characteristics. This is not certain, however, because the dialogue structures which have been implemented already do, in fact, take low confidence scores into account, using these to produce graceful degradation of the dialogue with the driver.

A **drawback** is that, apparently, we would need to be able to distinguish between different *causes* of lower-than-3 confidence scores or high numbers of user corrections and VICO repairs. Somehow, VICO must be able to verify that the recognition or NLU processing problems are due to a particular set of causes, such as accent, dialect, speech problems, difficult voice quality, language mastery deficiencies, etc. Without knowledge of those causes, the developed user modelling functionalities might help certain drivers much less than intended. So, the first question is if we can ascertain the causes behind the observed recognition and NLU problems. The second question, which may be just as serious, is whether we will be able to adapt VICO to adequately address each communication problem type.

The VICO recogniser does not appear to be able to distinguish among factors such as strong accent, strong dialect, and various speech disorders. All it can do about them at this point is to deliver sub-optimal confidence scores which might result from the presence of any of these factors, other factors altogether, or several of them in combination. An important question is how much this matters. What VICO can do in the presence of drivers who have these kinds of communication problems may be one and the same thing in all cases, namely, to modify its dialogue behaviour in ways which makes it easier for the recogniser and the NLU processor to deal with input from those drivers. One strategy for modifying dialogue behaviour to this end is to go into a more system-driven mode in which drivers have to answer specific questions most of the time, preferably questions for which short answers are sufficient. VICO already does that, in fact, as part of its graceful degradation strategy when facing low confidence scores in the input.

It would seem that the discussion above leads to a series of open questions which not only will need further discussion in the consortium but also empirical investigation of how the VICO recognisers manage input from drivers having the problems under consideration. Among other things, what we would like to know is:

- how do our current graceful degradation strategies manage communication with drivers who have strong accents or dialects, or speech disorders?
- given the elaborate machinery which VICO already has for managing various sources of recognition error, how reliable are low recogniser and/or NLU confidence scores and/or other measures, such as out-of-vocabulary word counts, for identifying that the problem is *either* one of a strong accent *or* one of a strong dialect *or* one of some speech disorder, *rather than* something else entirely, such as constant driver cross-talk with passengers in the car or excessive driver verbosity?
- if the problem under discussion is one which we have to do something about in addition to what is being done already, which among the currently feasible techniques are the most efficient ones? The answer may be that dialogue structure modification is the best approach in some cases but many other approaches might be considered depending on the exact nature of the problem.

Finally, it may be remarked that, beyond a certain point, there is probably nothing to do about very strong accents, dialects, or speech disorders, other than acknowledging that these drivers cannot profitably use the VICO system.

4.2.3 Information on the driver's communication attitudes

In Section 4.2.2 we discussed communication problems caused by relatively *permanent* driver characteristics. Other systematic communication problems, however, might be due to *more or less easily modifiable* driver characteristics, such as problems due to attitudes, habits, etc. which drivers have when they begin to use VICO but which VICO might be able to change more or less quickly. These communication problems include, among others, recognition problems and NLU processing problems due to:

- talkative drivers

- drivers talking to VICO and to passengers at (more or less) the same time, i.e. when VICO is listening
- drivers who are hesitant towards VICO because of the system's innovative or unfamiliar behaviour
- indecisive drivers
- overly polite drivers who keep addressing VICO with elaborate courtesy
- drivers who address VICO using an inappropriate speaking style, such as very slow and deliberate speech or very fast speech
- drivers who inadvertently tolerate regularly occurring disturbances from passengers and/or radio/music/TV/games

For instance, if the driver tends to produce too long utterances because of talkativeness or elaborate politeness, and if the NLU and/or the dialogue manager were able to tell that this were the case, VICO could store that information and use it to remind the driver to speak briefly and to the point. The same applies to inappropriate speaking style and toleration of external disturbance.

Some of the above problem types can be dealt with without the need for explicit user modelling. Thus, the VICO recogniser will be able to normalise with respect to slow and fast speaking style, using adaptation methods. This means that we can disregard slow/fast speaking style in our user modelling analysis. Similarly, radio playing will be captured by echo compensation, and the negative effects of background passenger voices will be reduced through using a microphone array with beamforming capability. So, in fact, the user modelling problems discussed in this section would seem to reduce to (a) how to handle talkative and overly polite drivers who produce lengthy spoken input, and (b) how to handle driver cross-talk to passengers whilst the recogniser is open. As for (c), hesitant and indecisive drivers, these will be discussed in Section 4.3 below because their problems would seem to be due first and foremost to lack of familiarity with VICO itself.

The VICO Daimler-Chrysler partner does not currently plan to produce speech recogniser detection of talkative drivers, overly polite drivers, and driver cross-talk to passengers, but is willing to look into these problems at a later stage in the project. These communication problems appear to be interestingly different from those discussed in Section 4.2.2, such as strong dialect or accent, speech problems, and awkward voice quality. In the latter cases, what VICO can do is to adapt its behaviour to the user. However, in cases such as the talkative or cross-talking driver, VICO should *not* adapt its general dialogue strategy. The reason is that it is not clear how this might help at all, given that the driver keeps generating serious difficulties for the recogniser and the NLU processor. Rather, the only practical solution may be that VICO should *try to change the driver's behaviour directly.* It seems that, compared to customisation and adaptation, this is a new type of service which VICO might provide the drivers. These services involve neither system adaptation nor system customisation but, rather, *user adaptation to the system at the request of the system*. System-requested user adaptation is similar to system adaptation to the user in that it is based on observation of the user's behaviour. Unlike system adaptation to users, however, it is the user who has to adapt to the system at the explicit request of the system.

System-requested user adaptation shares with system adaptation to users the **drawback** that the system will need to be able to distinguish between different causes of lower-than-3 confidence scores or high numbers of user corrections. Otherwise, the system cannot know which aspect of the driver's behaviour needs modification and hence cannot tell the driver what to do to improve

VICO's recognition and understanding of the driver's input. Clearly, we do not want to wrongly attribute a speech disorder to a talkative driver or to attribute talkativeness to a driver who has a strong dialect.

4.3 Information on the driver's experience with VICO

We expect, of course, that VICO will become an extremely useful tool for drivers, capable of offering an increasing number of services some of which are likely to be useful to virtually all drivers. This assumption generates another, i.e. that most drivers using VICO will become quite skilled users of the system, able to routinely carry out at least their favourite tasks without any problems. Still, drivers who are highly skilled in some of the tasks offered by VICO may remain less skilled, or even completely unskilled, in doing other tasks. Also, given the innovative nature of the VICO technology, it is perhaps unlikely that many drivers will have experience with similar systems the day they start as novice VICO users. This raises questions about what VICO might do for new users who are not familiar with the technology at all, and whether and how VICO might subsequently modify its behaviour vis-à-vis skilled users.

It should be pointed out that VICO is being designed and developed as a walk-up-and-use system in the sense that most drivers should be able to use VICO right away without having to resort to courses in how to use the system, comprehensive system manuals, etc. In particular, *how to operate VICO* should be a simple matter of speaking normally to the system and avoid verbosity. However, some drivers may not realise even that when they begin to communicate with VICO for the first time, as illustrated by some of the communication problem types discussed above. Moreover, exactly because of VICO's versatility, i.e. its ability to help drivers accomplish a range of different tasks, it will probably be much less evident to new drivers *what VICO can and cannot do* for them as compared to knowing or correctly guessing how to operate the VICO system. For instance, VICO is planned to have knowledge about 25-30 different types of point of interest, such as churches, and post offices. Few drivers, if any, are likely to remember the full list of types of point of interest which VICO knows about. This discussion leads to two key questions:

1. how will VICO know whether the present driver is a novice, intermediate, or expert in using the system?
2. given that VICO somehow knows this, what can VICO do to help all users except those who are experts in carrying out each and every VICO-supported task?

As to question (1), it is possible to build VICO user modelling functionality which enables the system to know, for instance, (i) if the driver is new to VICO and (ii) which VICO-supported tasks the driver performs regularly and which tasks the driver has not performed so far.

As for question (2), one solution could be to enable VICO to offer all new drivers general information on how to operate the system as well as on the tasks which VICO can help drivers solve. Similarly, drivers who are experts in some tasks but novices in others, could be offered task-specific information when they embark on tasks which they have not, or have not often, performed before. This solution, it seems, is straightforward and potentially very useful to all drivers. What it requires is, for instance, the development of a VICO information task module which can be called upon as required by the corresponding VICO user modelling functionality. The advantage of creating a separate VICO information task module is that drivers could call upon that module at any time and not just get access to information about VICO when the user modelling module judges this to be appropriate. This solution **does seem to meet our user modelling selection criteria** stated in Section 4 above.

Potentially, the solution just proposed could also be tailored to address the two points from Section 4.2.3 which were deferred to the present section, i.e.:

- drivers who are hesitant towards VICO because of the system's innovative or unfamiliar behaviour
- indecisive drivers

What this would take is to include in the VICO information task information which might help drivers unfamiliar with voice-operated technologies feel at ease when using them, information about timeouts and how they work, etc.

In fact, the VICO information module might also provide some amount of *preventive* support for avoiding the two other problems discussed in Section 4.2.3, i.e.:

- talkative and overly polite drivers who produce lengthy spoken input
- driver cross-talk to passengers whilst the recogniser is open

Again, this would be done by including information on the desirability of being brief and to the point when speaking to VICO, something which is already in the VICO information task as it stands, and to inform drivers that cross-talk with passengers while the recogniser is open is seriously counter-productive to getting the current VICO task done. It is not clear to which extent the VICO information module could *succeed* in preventing such problems. However, it seems clearly worthwhile to try to experiment with the proposed VICO information functionality before risking over-engineering by first investigating and then, possibly, building talkativeness, politeness, and cross-talk diagnostic competence into VICO followed by its on-line use.

At the time of writing, a first version of the VICO information task module has been implemented for English. However, further VICO information task development has been disrupted in order to focus VICO task development on the tasks stated in the VICO contract. Given what the present author considers to be the evident need for a VICO information task facility, a need which will grow rather than diminish as we keep adding new tasks to the system, it is hoped that the consortium can unlock this situation by focusing more on what drivers need than on what happens to be stated in the VICO contract.

4.4 Current user modelling priorities

Above, we have discussed at some length a series of issues and problems which VICO user modelling functionality might help with. In a brief summary, *information on the driver's task objectives* offers several types of information on drivers which may be considered user modelling candidates in the light of our selection criteria. *Information on the driver's communication with VICO,* on the other hand, does not at this point seem to offer any evident candidates for user modelling. We first need to conduct experiments and evaluations on the first VICO prototype in order to judge which among those kinds of user modelling functionality, if any, could meet our selection criteria. Finally, as regards *information on the driver's experience with the VICO system,* we have identified what appears to be an excellent candidate for user modelling, namely to develop and continuously update a VICO information task module which can be called upon by drivers at any time and which will also be linked to the VICO user modelling module. This module will then use the information provided by the VICO information module to help new drivers understand how to operate VICO as well as to understand what VICO can do for them, and help seasoned VICO users start to use VICO for tasks for which they did not use VICO in the past. In addition, the VICO information module will be able to provide some amount of preventive information to help drivers avoid communication behaviours which are counter-productive to getting their tasks done through spoken dialogue with VICO.

Whilst waiting for the consortium to decide whether to include VICO information functionality into the system, it seems that the user modelling functionality for the first VICO prototype will have to deal with some type of information on drivers' task objectives. In the light of the discussion in Section 4.1, we have chosen to demonstrate user modelling of drivers' hotel selection preferences in the first VICO prototype.

5. Task breakdown

This section discusses in more detail the VICO user modelling sub-tasks introduced in Section 2.

5.1 Identify the current driver

For user modelling to be of any use, VICO must be able to determine who among the car's drivers is currently driving. Moreover, this driver identification seems to have to be made with *near certainty.* If it is too uncertain that VICO has correctly identified the current driver, driver misidentification will happen too often. When this happens, some drivers will be "mistreated" by VICO because VICO communicates with them based on a wrong user model. Similarly, the information collected by VICO on misidentified drivers will be used to update other drivers' user models with the result that the user models for those drivers will become increasingly fudged up with irrelevant or distorting information. Thus, virtual certainty in driver identification is a must.

In addition, the driver must be *identified up front*, i.e. before or, at the latest, as soon as that driver starts the dialogue with VICO. If the driver is identified at any time later than that, the driver will be helped less by user modelling and each driver's user model will be missing important information on that driver's behaviour.

The above remarks lead to the question of *how to identify the driver up front.* There are several possibilities, at least:

- *voice identification.* Daimler-Chrysler has expressed their willingness to look into voice identification and they seem to believe that this would be feasible for VICO implementation. Even though today's voice identification technology is not perfect, it might be possible to get near-certain identification in the VICO case, simply because most cars have rather few drivers. Voice identification is also an elegant solution in the sense that the driver does not have to do anything other than speak to VICO about some task or other. It is not necessary for the driver to even explicitly address the identification issue. Even though VICO does not need data security at this point, voice identification does have the difficult-to-copy feature which might make it interesting should security considerations come up. Finally, voice identification can be made up front as soon as the driver speaks to VICO. A **drawback** is that VICO will not be able to respond by saying, as people would naturally do, e.g. "Hello Ole, what would you like to do today?", or just "Hello Ole" or "Ole". That would require VICO to be able to match the identified voice to the name Ole, which would require input keys or voice spelling, first-time drivers being requested by VICO to spell their names which VICO will then try to pronounce more or less appropriately. A **second drawback** is the reverse of the elegance of the pure voice identification solution, so to speak. It is that VICO will not be able to give the driver feedback that correct identification has taken place. To do so, VICO will have to associate the identified voice with some name, like in the example of spelling the driver's first name above, or with some code which the driver is told about;

- *driver's code.* "Password" may be a slightly misleading expression in this context since the primary purpose is not security but that of making the minimal distinction needed among the car's different drivers, so that, for instance, one driver is 1, another is 2, and so on. At some later stage in the development of VICO-style technology, though, it cannot be excluded that important elements of data security might become involved. In that case, a more elaborate driver's code could be an obvious candidate, at least as long as it does not have to be spoken to the system for all passengers to listen to. Contrary to the case of voice identification, a driver's code must be input to VICO explicitly. The simplest way to do so is to *speak(or spell) the code* to VICO up front. VICO might even start by asking for it, and VICO might then repeat the code as feedback to the driver that correct driver identification has taken place. Alternatively, the code could be entered through some other modality, such as *haptically* through one or several keys. This raises a new issue of adding key(s) to the VICO system, which has not been foreseen in the VICO workplan and which may complicate the VICO system unnecessarily, both technically and from a traffic security point of view. Simplicity and the absence of strong data security needs, therefore, speak for having an acoustic/linguistic code per driver. One **drawback** is that the driver will have to remember yet another code or password. A **second drawback** is that VICO will not be able to address the driver by first name unless code-matching first name spelling is added (cf. above);
- *driver's spelled first name.* This potential solution is different from the two options discussed above because now the driver must spell his or her first name *each time* the driver "logs on to" VICO. The spelled first name *is* the code for the driver, replacing the voice profile or some simple alpha-numerical code. This solution would enable VICO to give verifiable non-coded (first name) feedback to the driver that the driver has been correctly identified, simply by composing the word spelled and sending it through the synthesiser. A **drawback,** however, is that spelling one's full first name each time one wants to interact with VICO is an awkward thing to do, the more so the longer the name is. Drivers Elisabeth and Christopher, for instance, might not appreciate this solution. A "smart lexicon" in the VICO user model might solve this problem, so that drivers would only have to spell their full first name once. After that, it might suffice for them to spell only one or two letters for VICO to uniquely identify them among the first names of the car's several drivers. The only exception would be in families or, more generally, in driver groups for a particular car, in which one would find, e.g., a woman called Petra and a man called Peter. The worst case, of course, amounting to a **second drawback,** would be driver groups in which several drivers have the same first name. In such cases, whilst the first driver could register with VICO as, e.g., "Peter", the following Peters would have to invent variations on their names, thereby effectively approaching the code-spelling solution discussed above;
- several possible *combinations of the above* have been mentioned already, such as voice identification plus spelling of their first names by first-time users. This couples the elegance of voice identification with VICO's non-coded feedback on which driver VICO has identified. A perhaps slightly less elegant combination is a simple spoken driver's

code plus spelling of their first names by first-time users. Compared to the latter combination, the solution consisting in first name spelling supported by a smart lexicon is perhaps the simpler of the two, entirely avoiding arbitrary codes and having to remember these.

An additional point is the following. One should not underestimate the perceived trouble involved when drivers have to remember and to frequently speak codewords to VICO. To what may be only a slightly lesser extent, this remark also attaches to the use for spelling purposes of one or two first letters of one's first name. It is quite possible but by no means proven, that avoiding spelling or other explicit means of identifying themselves to VICO every time drivers have to use the system, could mean a major improvement in the perceived user-friendliness of the system. For instance, if the driver is assumed to spell a code or part of a first name but does not do so, as many drivers are likely to, VICO has to ask for the code or first name lest the user model of that driver will miss important information. This all contributes to creating a somewhat unwieldy user interface.

In conclusion, voice identification is the simpler and most elegant of the solutions discussed. If combined with once-and-for-all first name spelling for use in providing feedback to drivers, this solution may satisfy all the requirements which have been introduced above, including:

- virtual certainty of driver identification
- up-front identification of the driver
- no need to enter a code or (part of) a spelled first name at the start of each session
- immediate feedback to the driver on which driver VICO believes to be speaking to
- no new codes to remember
- VICO knowledge and use of drivers' first names to the extent that this is considered user-friendly dialogue design
- it is immediately clear if the driver is a new driver (see below)

The primary uncertainty remains the question whether it will be possible in VICO to achieve the required virtual certainty of driver identification by voice. If not, we will have to look for other solutions.

Referring now to the final bullet point above, in fact, voice identification, both with and without the once-and-for-all spelled first name option, has an additional advantage over all the other options considered. If a spoken code or (part of) a spelled first name is to be used for identifying the current driver to VICO, VICO can only access the driver's user model *once the spelling has taken place*. This means that it is only at this point that VICO will realise, for instance, that *this is an entirely new driver*. But how do new drivers know that they have to spell their names, or enter some spoken (spelled) code if they are not being told by VICO? The worst design solution to this problem is to make VICO assume, until proven otherwise, that the current driver is a new one who needs to be instructed in what new drivers have to do in order to make themselves identifiable by VICO. Regular VICO users will no doubt come to hate this interface very quickly. Moreover, it is not evident how to avoid rather user-unfriendly solutions to this problem, such as always first asking the driver if the driver is new to VICO. The best solution probably is to assume that the driver is new to VICO when the driver does not speak the code or part of a first name.

Another issue which has not been mentioned above, is that *passengers* might want to talk to VICO as well. Normally, it seems, a car has fewer different drivers than different, front seat or other, passengers. If the passengers also talk to VICO, and if VICO models all the passengers talking to the system, user modelling would become significantly more diverse than if only drivers are being modelled by VICO. VICO might come to have dozens of user models for a particular car, most of which are not being used at all since they were simply created when some passenger wanted to try to speak to VICO once. That is, we may probably safely assume that all or most passengers will only talk to VICO once or a few times. The question is whether this constitutes a problem or not.

Well, there are Problems and then there are (smaller) problems. From the point of view of engineering simplicity, it does not seem to be the most elegant solution to model any passenger who might only speak to VICO once. Still, assuming certainty of driver identification, the price to be paid in modelling the passengers as well simply is to get more user models. But these models will not fudge up the models of the "real" drivers. They just sit there in the user modelling module records and are of no use at all. The solution to be proposed here is just one among many. It relies on using any one of the double user identification approaches described above. Thus, any passenger can talk to VICO without a user model being created for this passenger, as long as that passenger does not spell a codeword, or a first name, to VICO. For instance, suppose that VICO will eventually do driver identification through voice. When a new passenger speaks to VICO, the same happens as when a new driver speaks to VICO, i.e. VICO discovers that the person is a new one and offers to that person to spell, once-and-for-all, that person's first name in order for a user model to be created of that person. If the passenger does not do this, no user model will be created for that passenger. In general, given the driver identification options discussed above, as long as passengers do not spell codewords and/or first names to VICO up front, they will not be modelled by VICO. They can speak to VICO as much as they want, of course, but no user model will be created of them. Some passengers may want to spell codewords and/or first names to VICO anyway, and that cannot be helped. In some cases, it may, in fact, be the passenger who does most or all of the talking with VICO, as in couples who drive together most of the time, the front seat passenger doing the talking with VICO. This may constitute a problem for in-car microphone placement but it does not constitute a user modelling problem.

Driver identification will not be implemented in the first VICO prototype in which we will only implement a single driver identifier which will be used throughout. Functionally, what needs to happen is that the driver's id is sent to the user modelling module which looks it up in its list of drivers ids. If there is a match, the driver's user model is retrieved (Section 5.2). If there is no match, a new user model is created for that driver (Section 5.3). The details of these processes as well as the accompanying dialogue with the driver will depend on the driver identification method which we will eventually adopt.

5.2 Retrieve the current driver's user model

Once VICO's user modelling module has identified the current driver as matching an already stored user model, this user model is defined as the active data structure and used for (i) supporting the driver during communication with VICO (Section 5.4), (ii) collecting driver information during the dialogue (Section 5.5), and (iii) updating the driver's user model with the collected information (Section 5.6). In future VICO implementations, the driver's user model may be retrieved over the CWW. At present, drivers' user models are located in the dialogue manager.

5.3 Create a new user model UM_Dx for a new driver, Dx

Once VICO's user modelling module has identified the current driver as a new driver by failing to match the current driver's id to those of the user models already stored, the user modelling module must create a new user model for that driver, label it, and add the label to its list of known user models. This process will be implemented for a single driver in the first VICO prototype. The new user model UM_Dx, where x is the label for the driver, is then defined as the active data structure and used for collecting driver information during the following dialogue.

5.4 Make appropriate use of the driver's user model during dialogue with VICO

By comparison with steps 5.1 through 5.3, the present step 5.4 which constitutes the actual driver adaptation phase, is likely to be considerably less simple. Its execution is likely to involve several dialogue manager modules, including the task manager, the user modelling module, and the domain handling module(s), if any, which is responsible for one or several particular tasks. The exact way to do this depends to a great extent on the kinds of information the user model stores on particular drivers (cf. section 3 and 4). In the first VICO prototype, the type of information used will be information on drivers' hotel selection preferences. We are presently refining the design specification for how this type of information will be handled by the relevant VICO dialogue manager modules.

5.5 Collect new information on the current driver from the driver's dialogue with VICO

At the same time, more or less, as VICO is using the relevant parts of its existing store of information about the driver for adaptive purposes (Section 5.4), VICO must also collect new information about the driver's behaviour during the ongoing dialogue. Assuming that VICO will be installed primarily in cars which have a small number of regular drivers, and assuming that most regular drivers will use at least part of the VICO functionality rather frequently, each driver's user model will come to include generalisations based on a considerable number of dialogue sessions with VICO. Given also the fact that a user model is supposed to be a (processed, cf. Section 5.6) record of past driver behaviour per task, we do not seem to need, in most cases, to collect driver behaviour information on-line as this behaviour is being exhibited. Thus, the simplest way to collect driver behaviour information would seem to be to use one or more of the VICO dialogue histories as the record(s) of reference for the information to be collected. This is what is being done with respect to the drivers' hotel selection preferences for the first VICO prototype. Once the driver is asked to provide the credit card number required for the purpose of hotel room booking, full information on the hotel to be booked by the driver is sent to the user modelling module.

It seems likely that the above approach of collecting driver information from the VICO dialogue histories at task (or even at session) completion time will be sufficient for many user modelling tasks. The distinction between task completion time and session completion time follows from the fact that a session with VICO may comprise the addressing of several tasks. Our current notion of session completion time is that this is when the car electrical system closes down.

Exceptions to the rule of collecting driver information at task (or session) completion time are, at least, (i) user modelling tasks which solely depend on the fact that the driver is new and unknown to VICO, and (ii) communication problems user modelling tasks for which a diagnosis and repair strategy can be developed and executed on-line without the need to await the problems summary delivered by the dialogue history at the end of a session with VICO. In these cases, user behaviour information would seem to have to be both collected and used on-line prior to the completion of any particular task. This should happen as soon as a new driver has been identified or a diagnosis made.

The VICO dialogue histories would seem to include almost all information on the driver's behaviour, including not only all of the driver's information choices but also confidence scores for each user input sentence. User-initiated and system-initiated meta-communication events are not marked up at this point but could be marked up if needed. As argued above, confidence scores and meta-communication events are good indicators of user-system communication problems but bad indicators of their causes. It would be desirable to be able to get more information from the speech recogniser and possibly from the NLU about the particular causes of particular communication problems in order to decide, at least, if VICO can do something about them or not. Such new event messages will then be handled by the dialogue manager.

5.6 Update the present driver's user model with the new information gathered

It is useful to distinguish between *when* to update the user model and *how* to update the user model.

5.6.1 When to update the user model

This issue has already been addressed in Section 5.5. The conclusion was that, for some user modelling tasks, user modelling update may safely be done at domain task or session completion time, whereas for other user modelling tasks, notably those where VICO can take action as soon as a new driver has been identified or a particular communication problem diagnosis made, user model updating must happen immediately. The latter approach takes into account the fact that unsuccessful or incomplete domain task dialogues are quite important for some user modelling purposes.

An issue to be kept in mind in connection with user model updating is that, in the cases in which updating is assumed to happen at session completion time, one or more *driver changes* may occur without the car's electrical system closing down. Thus, "session completion time" should perhaps be replaced by the more inclusive expression "session completion or driver change time". It is also quite conceivable that the driver may change in the middle of some task negotiation with VICO.

5.6.2 How to update the user model

How to update a user model involves three steps, at least:

1. receiving or retrieving the relevant information from the dialogue history or elsewhere (the task manager);
2. combining, per type of information retrieved, the retrieved information with the information already present in the user module for that particular driver and user modelling aspect. Depending on the type of information modelled, particular update algorithms may have to be developed for this purpose. For the hotel reservation task, for instance, we need a decision procedure for handling the driver who 10 times in a row chose a three-star hotel and who has now chosen a four-star one: is the default still the three-star hotel or something else?

3. storing the computed update results as an updated user model for that driver, the one which will be uploaded next time this particular driver starts using VICO.

5.7 Store the updated user model

In future VICO implementations, the driver's user model may be stored over the CWW. At present, drivers' user models are stored in the dialogue manager.

6. Embedded user modelling architecture

This section briefly touches upon a few major issues with respect to the user modelling module's architecture, role, and components. A general high-level specification of the module has been made in a separate document. We are presently diagramming the detailed architecture and information flow for the user modelling module.

6.1 Database

It is desirable to use a database for storing user information per driver and, for each driver, per user modelling task. A database makes it much simpler to scale up the driver information to include several user modelling tasks, store as comprehensive records for each driver as needed, have several types of record per driver/user modelling task pair, represent many drivers, as for use in rental cars, and enable fast database querying for instance in cases where the database might need to be consulted several times during a single task-oriented dialogue with the driver. Professional DBMSs are also easier to modify than home-grown data storage mechanisms.

6.2 Modularity

It is important to design the user modelling module in a modular fashion, so that it becomes easy to not only add new driver models for some user modelling task but also new models for new tasks. Similarly, no commitments should be made as to the kinds of data structures which could be stored in, and retrieved from, the database, since these data structures may turn out to be considerably different for different user modelling tasks. Also, the user modelling module controller must be designed in such a fashion that the controller can easily be made to control additional user modelling tasks.

6.3 Where to put the user models

One aspect of the user modelling module architecture which is of particular interest to the VICO partners is whether to:

- put drivers' user models on the CWW and keep them there for querying and updating;
- put drivers' user models on the CWW and upload the user model of the current driver UM_Dx to the dialogue manager during the drive for on-line consultation and updating followed by storage on the CWW at the end of the drive or when the car's electrical system closes down, or when the driver changes;
- put drivers' user models into the dialogue manager and keep them there.

All three solutions may be possible in principle. The third and last option, which is the one adopted for the first VICO prototype assumes that information can stay permanently in the dialogue manager also when the car's electrical system is closed down.

Acknowledgements

Thanks are due to André Berton and Fritz Class from Daimler-Chrysler for their comments on what may be modelled and what does not need to be modelled concerning the driver's communication with VICO. Thanks are also due to Laila Dybkjær for her comments on an earlier version of the present document which formed the background for the driver identification and user modelling sections in Bernsen, N. O. and Dybkjær, L.: A multimodal virtual co-driver's problems with the driver. In *Proceedings of the ISCA Tutorial and Research Workshop (ITRW) on Spoken Dialogue in Mobile Environments,* Kloster Irsee, Germany, 2002. Finally, thanks are due to Petra Geutner who provided constructive questions on an earlier draft.