Navigation System: An Experiment

T. Nestorovič
University of West Bohemia in Pilsen, Czech Republic, Email: nestorov@kiv.zcu.cz

Introduction

Modern car navigation system is a great benefit for a driver. Not only it cuts down his/her load during the drive, but moreover, it adapts to dynamically changing conditions, as for instance blocks or driver's incoherence with trace computed. However, the counterpoise of all pluses seems to be the user interface itself – whenever the driver needs some information, s/he is forced to look at and/or manipulate with the system touch screen. If it is not located in the road eye angle of view, it rapidly distracts the driver's attention. Accident statistics prove, that a half of car crashes is caused by looking at system display while driving [1]. A very probable solution to this issue seems to be to get speech interaction involved. For this way of information exchange to be present in navigation system, it is necessary to transform current graphical interface to a voice-controlled one and design a simple (as far as possible) form of dialog for (almost) each possible function of a particular navigation system. Although some of car manufacturers decided to adopt this approach of controlling, speaking systems are still a feature of more expensive car models only.

This article is conceived in an experimental navigation system description. The topics are as follows. First, the overall system architecture and its spoken dialogue management capabilities are presented. Then, the user interface proposed is described, including general rules for a selection of those system functions which are candidates for multimodal approach. Finally, an experiment conducted on *N*=11 volunteers is described and currently undertaking improvements of the navigation systems mentioned.

Navigation System Description

Architecture Overview

The system was from its early beginnings designed as multi-modal due to two reasons. First, not all graphical functions of a common navigation system are easy to convert to spoken counterparts (see below the three general rules). Second, spoken interaction is a priori error prone, and even more so if the speech recognition is applied to a car environment.

The overall architecture of the navigation system is depicted in **Figure 1**. The user (driver) can interact with the system using either spoken language, or touch screen. For the purpose of the experiment, spoken language was substituted by a text interface and the touch screen by a Flash movie[1] content. However, none of these substitutions invalidates the results obtained.

The core component of the system is the dialogue manager. Which one of two modalities to choose (and why) is, for example, one of its load (more detailed capabilities description follows). It plays not only the role of the system "brain" module, but it maintains the domain knowledge as well – a strictly centralized approach was chosen (all component-dependent information are stored here), a lucidity property was the reason.

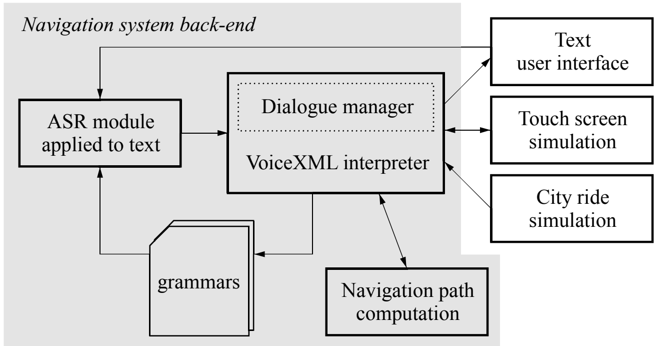

Figure 1: Presented navigation system architecture – back-end and front-end components.

Dialogue Manager Capabilities

The manager is designed as domain-independent and easy-to-extend. It deals with flat frames, thus is intended for information retrieval simple tasks only. All particular dialogues (information flow as well as elicitation strategy) are modelled by calling basic functions provided by the manager – more complex functions (domain-*dependent*) are based upon these ones. The description of a dialogue behaviour is called a *scenario*. Additionally, the manager maintains a history of interaction, and therefore, dialogues can exhibit a simple intelligence.

Scenarios are given a number expressing their urgency of accomplishment. For example in our navigation system the scenario describing dialogue when a phone call is received has higher urgency level than a scenario describing an interaction of saving an address into a directory. Hence, when the driver discusses which address to save and a call comes, the current task is suspended and the incoming call shifts to the focus of interest.

The manager also provides four dialogue flow modes, distinguished from each other by how much initiative is given to the user within a dialogue. The modes are *non-restrictive* (user may utter any sentence within the domain), *semi-restrictive* (user's initiation is restricted to a current task only), *restrictive* (user is allowed to answer the current system question only) and *alternative*[2] (the user is redirected to an alternative input modality). Whereas the restrictive mode serves as a state-based dialogue management approach

1 Adobe Flash; the term "movie" is a standardized terminological expression standing from the early age of Flash technology.

2 This mode is undefined (as it is domain-dependent property) and the manager expects it will be either defined, or disabled.

(turned to when the user seems to be unable to provide his/her demands properly), the semi- and non-restrictive modes are intended as mixed- and user-initiative approach, respectively.

Along with the above modes, the manager provides a basic mode selection strategy built upon how well does the dialogue progress. The basic strategy may be overridden by the domain designer. However on the basis of a current mode, the manager produces the grammar which is fed in the next step into the ASR module (Automatic Speech Recognition).

The manager produces TTS output (Text To Speech). This seems as a drawback, because, compared to CTS (Concept To Speech), no human-like artefacts (ellipsis, etc.) can be injected in the output. To the topic of this manager improvements see the "Future work" section at the end.

Navigation System User Interface

As mentioned above, the user interface (UI) must have been designed multi-modally, because of ASR quality constrains in general – the ASR module is the weakest part of every spoken language system [2, **7**]. Moreover, in the navigation domain in particular, not all functions were (could have been) designed multi-modally. A function has a spoken alternative if it satisfies at once:

- the driver uses it often while driving (telephone functions, or next direction question),
- it facilitates the manipulation with the system (a counter-example may be building an itinerary – easy to do graphically, but hard to accomplish using speech), and
- its results are easy to present using speech (if not, it makes no sense to provide a spoken alternative – the driver will anyway need to look at the display to get the results).

The development of the UI was conducted by these three rules. The navigation system [3] served as a basis of functions expected in this domain, however, not all of them are implemented here. The focus was put mainly on functions related to navigation itself and board phone controlling. The voice menu proposed is in accordance with classical GUI structure. The reason for this decision was to

- meet the coherence of GUI and SUI menu structures, because
- users tend to adopt system's terminology, or express themselves in a way they learned in the past [2, 8]; moreover, people like known thinks, and use of GUI-like *metaphors* [4] seems to be a good idea.

Therefore, conceiving the SUI menu structure in an unique way (i.e., differently), the system overall confidence and user-friendliness might at stake – people dislike and try to avoid inconsistency [5]. The menu structures for both modalities are, therefore, the same as shown in **Figure 2**.

The driver's movement within the menu structure is affected by the dialogue manager current mode. For example, in the restrictive mode it is necessary for a particular sub-menu from the Main menu to be visited before a desired function can be invoked. Another constraint is put on *when* the manager forces the driver to utter. A very simple approach is employed: it happens when discussing a function only – in any of menus, the driver may keep silent. A more complicated approach would regard a traffic situation as well – for example, when the driver is preparing to a difficult maneuver, the system better limits its dialogue management demands.

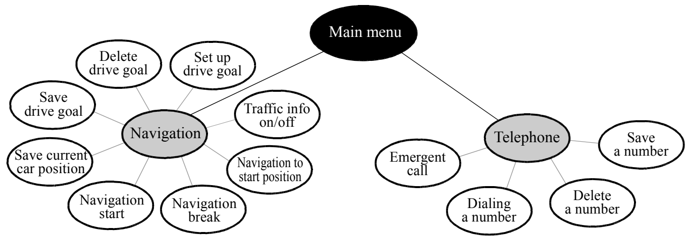

Figure 2: The menu structure is designed the same way for GUI as well as for SUI, consistency pursuit is the reason.

Navigation System Experiment

Experiment Description

The goal of the experiment was first to decide whether the navigation system UI proposed provides enough freedom and robustness, and second, to test the dialogue manager's management capabilities (uni- as well as multi-modal). The test was conducted on N=11 volunteers and it had the scenario of a "journey to the grandmother." The participants were instructed about the UI,[3] and then given three tasks to accomplish, all of them were evaluated (on the basis of [3]) as the most typical/important ones the driver may encounter during the drive:

- starting the navigation,
- dialing a particular telephone number, and
- saving the car current position.

To start a navigation, the user needed to utter a particular demand[4] and provide an address (s/he was informed in advance which one is the correct one). When an address is needed, the system always temporarily redirects to the touch screen. After the address is cleared, the optimal path search criterion is asked (i.e., preferring highways, avoiding roundabouts, etc.). The function also discusses the situation when another navigation process is running; however, this condition was never met.

To dial a particular telephone number, the user needs to know either the number itself (the case of this experiment), or a shortcut to an inner list of stored numbers. When the phone call is over and the number called is not stored in this

3 Information provided was on how to use and manipulate the UI. They also were called attention to the possibility of system's right to change the modality when the dialogue begins to stagnate (i.e., seems to not lead to a successful end). Additionally, volunteers were asked to not correct mistypes in their utterances – the reason was at least somehow to emulate ASR errors.

4 I.e., "I wanna be navigated to my grandma," or finding the correct function by walking through the menu hierarchy.

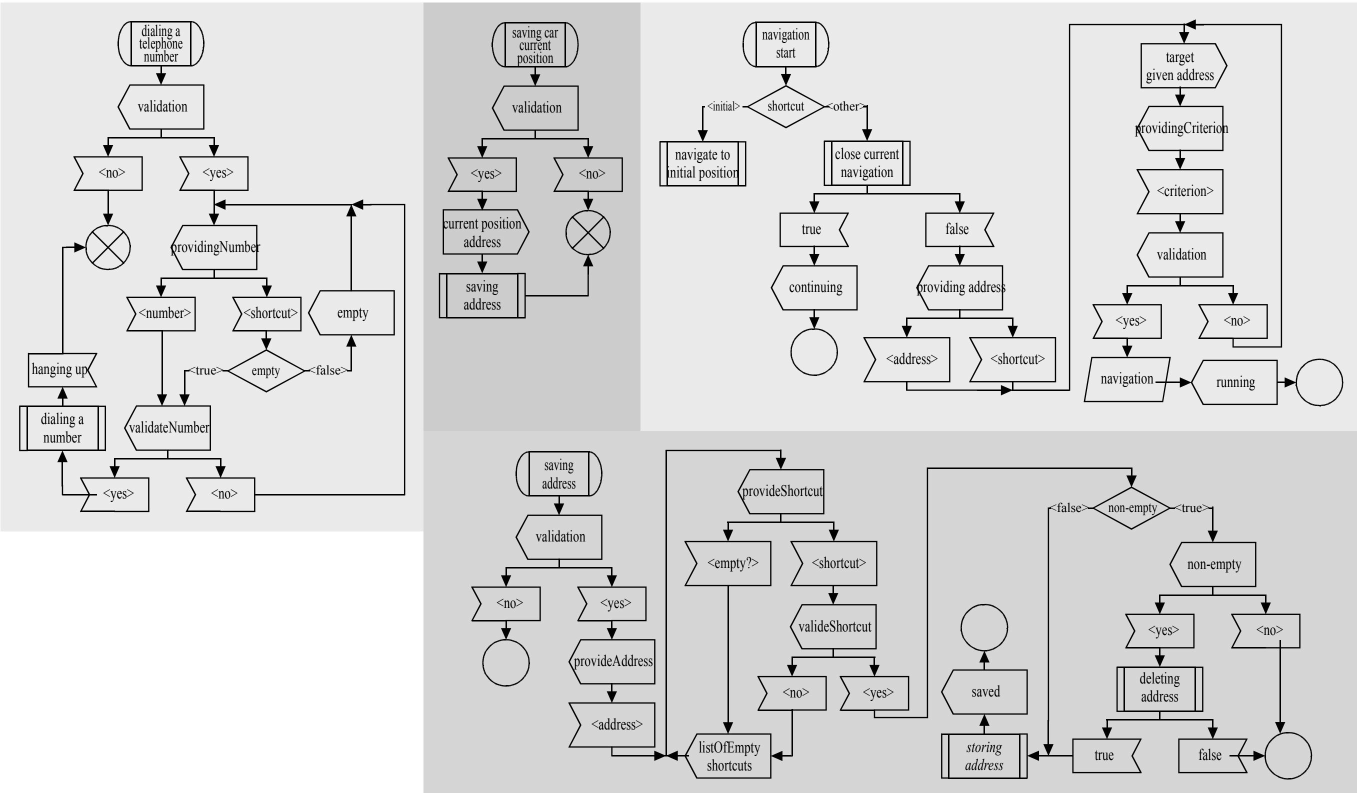

Figure 4: The SDL notations for “Dialing a phone number”, “Saving car current position”/”Saving address”, and “Navigation start” tasks.

list, the system asks whether the user wants to save it, or not and invokes another task accordingly. In this new task, only a shortcut needs to be provided and validated.

Finally, to save the car current position, the user may either find appropriate function in the menu (needs only to distinguish a shortcut in a list of addresses and have it validated), or undertake a selecting of the “current position (i.e., address)” in a process similar to starting a navigation augmented with providing a shortcut the address will be stored under.

The complete formal description of these three tasks is depicted in **Figure 4**, using the SDL notation [6].

Results Obtained

Test parameters tracked were measured for each task separately: the number of turns[5] and clicks, and the amount of time needed to successfully accomplish the given task (despite the resolution of times estimated in advance – in common UI's, the graphical always manages to beat the spoken one – the estimation proved to be true, see **Table 1**).

There was also a lot of common human-computer interaction paradigms observed. Especially in the beginning of a session, volunteers tended to overestimate system's capabilities providing too complex demands (“How do I get to Prague?” - although the system comprehends it is asked to start a new navigation, it is unable to recognize the city provided, too large variability is the reason). However, after a certain period of time, participants adjusted their utterances by shortening them (**Figure 3**) and tended to be led by the system rather than exhibit own initiation. When performing the third task (saving the car current position), some of participants did not listen to the system prompt1 until its end and managed to accomplish this task using the longer solution explicitly providing the “current position.”

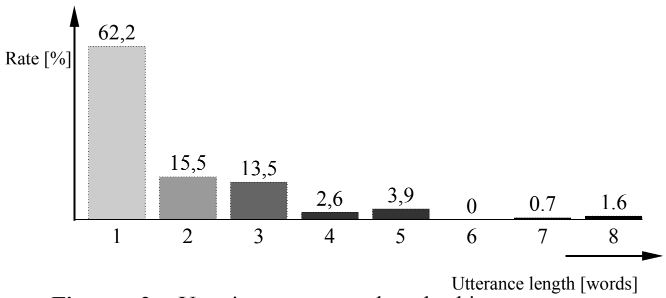

Figure 3: Users' utterances length histogram; users expressed themselves vastly briefly and did not exhibit own initiation.

In two cases, the dialogue manager gathered from user's utterances that the spoken dialogue is an improper modality and correctly redirected it to the touch screen (**Figure 5**).

One of the first task dialogues was as follows. Note, that in this case, the user was able to formulate his demands, and the redirection in the dialogue is, therefore, only temporal.

U I wanna go to Prague.
S Do you want to input the target address and start the navigation?

(the dialogue continues on the next page)

5 A *turn* is perceived as one iteration through the “system question – user answer” loop.

U Yes, I do.
S Switching over to the touch screen. Please select an address on it.
U *[Selects an* ADDRESS *and presses the Ok button]*
S Switching back to the non-restricted voice mode. Choose a criterion for navigation to *ADDRESS*.
U I want to get there as quickly as possible.
S Do you want me to find the fastest route?
U Yes.
S Route found, starting the navigation. Drive straight. You are in the main menu.

	SUI			GUI	
	Time [s]	Turns [-]	Clicks [-]	Time [s]	Clicks [-]
Task #1	143,7	6,6	4,0	33,9	11,3
Task #2	66,3	6,7	0,0	24,7	7,1
Task #3	75,3	9,0	0,4	18,9	5,9

Table 1: Numerical results of the experiment.

Figure 5: An example of touch screen display state. Here, the user is asked to provide a destination to navigate to.

Future Work

Currently, the main effort is directed to improve the dialogue manager. Its inner structure is represented using nested frames, thus enabling more and better structured domain segmentation. It is expected to have a perceptible impact on the structure of domain language model – shortening of phrases. For example, currently, to save the car current position, it is necessary to utter a sentence containing "Save *current* position"-like phrase; having the domain model better structured should enable sequential narrowing of user's demand, i.e. "Save" followed by "Position" followed by "The current one" etc.

As soon as a certain level of the manager's completeness is reached, it is planned to continue with the research within this domain and transmit the navigation system into the new version of the manager. Under better dialogue management circumstances, it is expected the results to be decently better, i.e., more smoother communication between the system and its user.

Conclusion

This article presented our approach to a multi-modal car navigation system. The architecture and dialogue management capabilities were presented as well as results in brief. On the basis of our results, the users were able to accomplish all given tasks. However, some improvements are still to be done, including clarity of some system prompts.

References

[1] Matoušek V., Nestorovič T. "Návrh hlasové komunikace s navigačním systémem automobilu a její implementace v jazyce VoiceXML." In *Proc. of NavAge*, Prague, 2006.

[2] Gustafson J., *Developing Multimodal Spoken Dialogue Systems - Empirical Studies of Spoken Human-Computer Interaction*, Ph.D. thesis, KTH, Department of Speech, Music and Hearing, 2002.

[3] BMW AG, "Bordmonitor mit Navigationsystem und TV."

[4] Dorey S. J, "Common User Access – Principles of GUI Design." 2007. URL: `http://www.susandoreydesigns.com/software/CommonUserAccessGUIDesign.pdf`

[5] Nass C., Ulla G. F., Somoza M. "The Effect of Emotion of Voice in Synthesized and Recorded Speech." URL: `http://www.stanford.edu/~nass/comm369/pdf/VoiceandEmotion.pdf`

[6] Nyeng A., "The CCITT Specification and Description Language – SDL," URL: `http://www.telenor.com/telektronikk/volumes/pdf/2_3.1993/Page_067-070.pdf`

[7] Yankelovich N., "How do users know what to say?" In *Interactions*, vol 3, 1996, pp. 32-43.

[8] Hobart J., "Principals of Good GUI Design," Classic System Solutions. October 1995. URL: `http://www.classicsys.com/css06/cfm/article.cfm?articleid=20`