Adaptation of an Automotive Dialogue System to Users' Expertise

Liza Hassel[*+] and **Eli Hagen**[+]
{liza.hassel, eli.hagen}@bmw.de

[*] Centre for Information
and Language Processing
Ludwig Maximilian University
Munich

[+] Forschungs- und
Innovationszentrum
BMW AG
Munich

Abstract

Spoken dialogue systems (SDSs) can be used to operate devices, e.g. in the automotive environment. People using these systems usually have different levels of experience. However, most systems do not take this into account. In this paper we present a method to build a dialogue system in an automotive environment that adapts to the user's experience with the system. We implemented the adaptation in a prototype and carried out exhaustive tests. Our usability tests show that adaptation increases both user performance and user satisfaction.

1 Introduction

Since the first in-car SDSs appeared in the late nineties (Heisterkamp (2001); Haller (2003)), a lot of research has been carried out to make these systems adaptable: Cnossen et al. (2004) and Piechulla et al. (2003) investigated the task demand on user attention; Mourant et al. (2001) did research on the influence of age on user behaviour; Rogers et al. (2000) investigated the adaptation of in-car information services to user preferences; the main research focus of Akyol et al. (2001) and Libuda (2001) was on clarification dialogues and how to deal with errors. In these studies emphasis was on other factors than the user's knowledge about the system, thus we chose that as our basis for adapting the dialogue system's behaviour.

A dialogue system that behaves in the same way for all users, in spite of their different degree of experience, will, for neither of them, be a truly usable interface (Wu, 2000). Users who have little or no experience with a dialogue system will have different priorities, expectations and needs than experienced ones. Because novices know little about a system, they will need a kind of tutorial to learn how it works. A further help for these users should consist of detailed confirmation of the actions triggered by their voice commands. Experts will want to accomplish the desired tasks quickly and straightforward. Such users need little guidance and the confirmation prompts should be short or could be left out completely. A solution to this dilemma is often the development of systems that are very easy for novices, but not very effective thereafter (Landauer, 1997). Ideally, the system should be able to recognize the degree of expertise of the user and adapt to it (Nielsen, 1993). Only in such a way it can provide at all times and for all users a maximum performance and user satisfaction.

Based on these observations, we developed a system that adapts its prompts as users become more experienced. The adaptation is done at the textual level, i.e. exclusively the information content of the system prompts is adapted, but neither dialogue strategy nor functional range.

Our speech interface was implemented as part of BMW's iDrive system (Haller, 2003). In addition to speech, iDrive has a manual-visual interface with a central input device in the centre console (controller, see figure 1) and a central display in the centre column (see figure 2). When users operate the controller (turn left and right, push in four directions and press down), they receive visual feedback on the display.

Figure 1: Controller and PTT-button

Over the speech channel, users can operate tasks in the areas entertainment, navigation, communication and climate control. Users activate the ASR with a push-to-talk (PTT) button on the steering wheel or in the middle console near the controller. The dialogue style is command and control as illustrated in tables 1 and 3 (p. 4). Users

Figure 2: Display Control

can ask for help (general characteristics of the SDS) or for options (in the current dialogue state available voice commands). In the prototype we have implemented additional. They are discussed in Hassel and Hagen (2005)

Novice	Expert
user: Entertainment. system: Entertainment. You can say AM, FM, or CD.	user: Entertainment. system: Entertainment.
user: Choose CD. system: Say a CD number	user: Choose CD. system: Number?
user: *Unintelligible*. system: I could not understand you, repeat.	user: *Unintelligible*. system: Pardon me?

Table 1: Novice and Expert Prompts

The iDrive SDS is currently configured for ca. 3000 words and phrases. For our experiments, we used the German version. In Hassel and Hagen (2005) we describe the experiments in detail. For further information about iDrive see Hagen et al. (2004).

In section 2, we introduce our method for classifying users. We then explained the proposed adaptation (section 3) and report the effects on usability (section 4).

2 Classification of the Users

The overall goal for our work is to increase user satisfaction and efficiency with the speech interface. One vehicle to achieve this goal is to account for the user's experience with the SDSs. We assume that experts need and want less guidance than novices. In a SDS, this aspect can easily be conveyed at the textual level, as table 1 shows. Currently, we are using only the two categories, novice and expert, but the concept easily allows for the inclusion of intermediate ones. Future work will address the optimal number of categories needed.

2.1 Adaptation Concept

Adaptation should happen unnoticed in the background without interfering with the problem solving task (and the driving!) (Rogers et al., 2000). We developed a concept that unobtrusively analyses the user's behaviour while interacting with the system. After every interaction, the system assigns users a category in a range between novice and expert. This categorisation accounts for every task separately, i.e. users can be experts at telephone tasks (e.g. dialing a telephone number) but novices at navigation tasks (e.g. changing the map). In this way, the system can account not only for different global levels of experience, but also for different patterns of use. For the calculation of the user category for a given task, we calculate a user model in terms of a vector $\vec{UM}$ using the following parameters:

Parameter	Meaning
# help requests, h	Users asked for general information about the system
# option requests, o	Users asked for the currently available voice commands
# timeouts, t	The system did not get any acoustic signal
# ASR-failure, e	The system could not understand the users input, e.g. OOV words or unintelligible speech
Relative use freq., $H_r = \frac{\|task\ t\|}{\|interactions\|}$	How often users activated task t in relation to all other performed tasks
Mean response time, T_m	How long users needed to answer, also known as onset time
Confidence measure, K	How reliable the speech recognition was

The user model $\vec{UM}$ is updated after every interaction, and it is the basis for calculating the user's current position on the expertise scale. The process is illustrated in figure 3. $\vec{UM}$ is multiplied with a weight vector $\vec{UM}_{ref}$ representing the importance of each component of $\vec{UM}$. The scalar product Δ_{UM} is compared to a threshold value s. If $\Delta_{UM} > s$, users behave as novices, if $\Delta_{UM} \leq s$, they behave as experts.

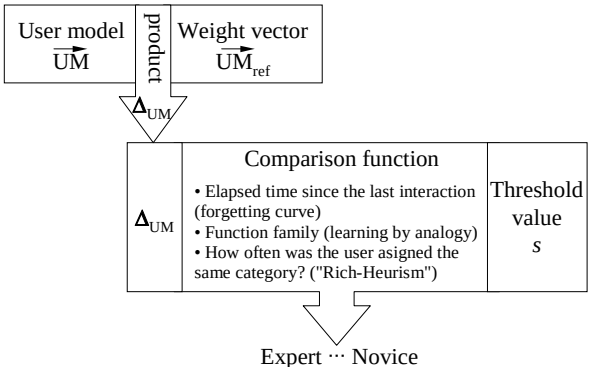

Figure 3: Calculation of the User Category

A user is categorised for the actual task. When he/she calls this task again, the system responds with the appropriate prompt - for novice or expert. Table 1 shows three examples for system responses for experts and novices.

The comparison function takes three values as input: Δ_{UM} (see above), s, and chance behaviour. The threshold value s accounts for people's ability to learn by analogy (section 2.2) and for the fact that people tend to forget new acquired knowledge quickly (section 2.3). A heurism makes sure that wrong category assignments due entirely to chance behaviour are avoided (section 2.4).

2.2 Task Families

Our domain consists of many similar tasks (e.g., the command "play" in the CD and DVD menu), thus we include peoples's ability to learn by analogy into our modification of s. We assume that when users know how a particular task works, they will learn the use of a similar task faster and in an easier way than the use of a completely different task. Thus, we have grouped the system tasks in families, according to their similarity. Table 2 shows an example for a possible task family, *setMusic*.

Speech Command	Family
play	setMusic
scan	setMusic
repeat	setMusic
random	setMusic

Table 2: The *setMusic* Family in the Entertainment Menu

Users are classified as novice or expert for each task in a family. When a user calls a task of a family, the category assignment depends on the status of the other tasks in this family, i.e. if a user is already classified as experts for other tasks in that family, then s is reduced. If users are classified as experts for more than 50% of the tasks of that family, they are then classified as experts for all tasks in that family.

2.3 Elapsed Time since the Last Interaction

People tend to forget newly acquired knowledge rapidly, unless they revise it frequently, until it becomes part of long term memory (Edelmann, 1996). The influence of these psychological insights on the learning behaviour of the users is incorporated into the adaptation concept by adapting the threshold value s to the elapsed time between interactions. The more time elapses since the last interaction the faster users are assigned novice status, i.e. s is reduced.

The real value of this factor could not be evaluated since our tests only lasted 45 min. We need long term evaluations to elicit the time after which we can assume users to have forgotten the voice commands. The alternating forgetting and learning periods give rise to a forgetting behaviour that can be described with a differential equation. The reason is that the intervals during which people forget a certain percentage of the commands become longer over the time because the acquired knowledge becomes gradually part of the long term memory. One should bear in mind these reflections when evaluating the elapsed time factor.

2.4 Chance Behaviour

We included a heurism in order to account for user behaviour produced by chance: After a prediction of the user model, look for positive/negative verification for this prediction in the next interaction (Rich, 1979). We fulfil this claim as follows: The comparison function has to yield three times the same result for a certain task before users are assigned another category. That is, if users have novice status for the task t, they have to behave three times as experts before the system assigns them expert status for t. Doing so, the probability that the system categorizes users in a category only by chance is minimized.

Although we could not colletc empirical data for the use of a threefold cycle, the results of our tests indicate that 3 is a good choice. 41% of the reference test subjects (TS_A) agreed with the statement "lists with options are too long", but only 10% of the prototype users (TS_B) did. TS_A had to ask explicitly for options; TS_B (with novice status) were told the options wihtout asking for them. Bearing in mind this difference and the answers mentioned above, we conclude that, at the beginning, the repetition of the options is necessary and that the chosen three times is not too much.

3 Adaptation of the System Prompts

The difference between system prompts for novices and for experts is mainly their explicitness, e.g. while for novices the SDS mentions the available voice commands without waiting for users to ask, experts have to explicitly ask for them (see table 3). Long and informative prompts would be in the long run annoying to frequent users.

The different system utterances were analysed with respect to the information they convey to users and assigned a DAMSL-category (Core and Allen (1997); Allen and Core (1997)). Depending on the semantics they transmit, the information can be presented in different ways (cf. table 3). To this end, we rely on the notion of Grice's *conversational implicatures*, and the basic principles for their calculation: cooperation principle and conversational maxims (Clark, 1997).

Contributions conveying conventional meaning do not necessarily need linguistic signals. For experts, Openings and Closings can be performed by tones, and a Signal-understanding confirming an action requirement like "play CD" can indirectly be accomplished by playing the CD. Contrarily, Signal-understanding confirming a dialogue state transition, e.g. "entertainment", needs linguistic signals to express the confirmation. For novices, these utterances can be completed with the avail-

Utterance Type	Novice	Expert
Opening/Closing	user: <PTT> *(Action-directive)* system: Speech input <Tone A> / Speech input terminated <Tone B>	user: <PTT> system: <Tone A> / <Tone B>
Signal-understanding	user: Play CD. *(Action-directive)* system: CD is being played.	user: Play CD. system: <Music is heard>
Signal-understanding (+ Open-option)	user: Entertainment. *(Action-directive)* system: Entertainment. Say AM, FM, CD or DVD.	user: Entertainment. system: Entertainment.
Assert	user: Destination input. *(Action-directive)* system: This task is currently not available.	user: Destination input. system: Currently not available.
(Assert +) Action-directive	user: Select CD. *(Action-directive)* system: CD slot is empty. Insert a CD.	user: Select CD. system: Insert CD.
(Signal-non-understanding +) Action-directive	user: <Not recognised> system: I could not understand you, repeat.	user: <Not recognised> system: Pardon me?
Signal-non-understanding + Open-option	user: <Not recognised> system: For general information say help, for available commands say options.	user: <Not recognised> system: You can ask for help or options.

Table 3: Classification of the System Utterances: Examples

able voice commands to help the user to carry on with the dialogue.

An assertion cannot be completely replaced by non-linguistic signals. This kind of prompt can be expressed in a less verbose manner or, at the most, be inferred from another prompt type through implicature. A example for the use of implicature in expert prompts is the combination of Assert and Action-directive, e.g. users can deduce from the directive "Insert a CD" the assertion "'CD slot is empty"'.

Signal-non-understandings could also be replaced by a non-linguistic signal. However, a beep may not fulfil the maxim of quantity (= make your contribution as informative as is required), since a tone alone may not be enough to indicate users what to do next. Besides, it may not fulfil the maxim of manner (= avoid obscurity of expression and ambiguity) because it might be difficult for the driver to discern between Signal-non-understanding and Signal-understanding tones. Therefore, we decided to express non-understanding using utterances like an Action-directive, e.g. asking users to repeat the last utterance.

Table 3 summarizes these examples. The system prompts type is given in the first column. Prompt types set in brackets were left out for expert prompts. The type of the user's utterances is set in brackets.

4 Conclusion

In this paper we described the classification of users between beginner and expert, and the adaptation of the system prompts to the calculated user expertise. The adaptation was assessed in a real driving situation in two test series (reference system and prototype). The evaluation showed that adaptation contributes significantly to enhance the usability of the SDS.

The comparison of prototype and reference system showed that adaptation contributed to improve the usability. Users of the prototype needed both less time and turns to complete the tasks than users of the reference system. TS_A needed on average 62.1 sec to complete a task, and TS_B 47.0 sec. TS_A needed on average 8.7 turns to complete a task, and TS_B 6.9 turns. Furthermore, 77% of the TS_B declared that options should be prompted after every system utterance, at least at the beginning, but only 27% of the TS_A agreed with that. While users that did not try adaptation were sceptic about it, the ones that tested adaptation wanted to have it afterwards.

User satisfaction and task success provided further evidence in favour of adaptation. User satisfaction was higher for the prototype, and while TS_B could complete 94% of the tasks, TS_A could complete only 81% of them. Moreover, TS_B requested options only $\frac{1}{5}$ of the times TS_A did. In general, users found the enumeration of the available options a good means to learn the systems, but in the long run the enumeration would be tedious. Therefore and because they knew they could asked for options and help, TS_B approved of adaptation.

In automotive environment it is important that users keep their eyes on the road. 68% of the TS_B did (almost) not look at the display, in contrast to only 45% of the TS_A. This can be interpreted as a sign that TS_B knew what to say without having to look at the display, and that they realized how comfortable it is to have the available voice commands read, at least until they are known.

According to Nielsen (1993), systems designed for novices should be easy to learn, i.e. the learning curve should be very steep at the beginning. Our test showed that TS_B reached soon the asymptote of the curve be-

cause they learned very fast how to use our prototype. The system prompts for novices served their purpose. We could confirme that the initial part of the learning curve for the prototype's users corresponds to the recommended shape.

Systems designed for experts are hard to learn but highly efficient, i.e. the learning curve is even at the beginning (Nielsen, 1993). TS_A learned by trial and error that they can speak the tasks they want to activate directly, leaving out the nodes between, that is, the reference system fullfils this claim. The next question is if our prototype would also fulfill the requirements stated by Nielsen (1993) for experts. The prompts of system B for experts turn quite the same as those of system A, this improves the efficiency. Furthermore, the prototype offers users a "suggestion" feature (Hassel and Hagen, 2005) to learn better ways of completing a task. Long term experiments still have to show if system B displays a typical expert learning curve over the time. Besides, further tests have to be done to confirm these assumptions also for elderly people.

Aknowledgements

We thank Klaus Schulz (LMU, Munich) for helpful discussions clarifying our ideas and for comments on earlier drafts. We also thank Stefan Pöhn (Berner & Mattner) for the programming, helping to make our, often chaotic, ideas concrete. Thanks to Alexander Huber (BMW AG) for his continuing encouraging support.

References

S. Akyol, L. Libuda, and K.-F. Kraiss. 2001. Multimodale Benutzung adaptiver Kfz-Bordsysteme. In Thomas Jürgensohn and Karl-Peter Timpe, editors, *Kraftfahrzeugführung*, pages 137–154. Springer-Verlag, Berlin.

J. F. Allen and M. G. Core. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. http://www.cs.rochester.edu/research/cisd/resources/damsl/.

H. H. Clark. 1997. *Using Language*. Cambridge University Press, Cambridge, New York, Melbourne.

F. Cnossen, T. Meijman, and T. Rothengatter. 2004. Adaptive Strategy Changes as a Function of Task Demands: A Study of Car Drivers. *Ergonomics*, 47(2):218–236.

M. G. Core and J. F. Allen. 1997. Coding Dialogs with the DAMSL Annotation Scheme. In *AAAI Fall 1997 Symposium on Communicative Action in Humans and Machines*, pages 28–35. American Association for Artificial Intelligence (AAAI). URL: http://citeseer.nj.nec.com/core97coding.html.

W. Edelmann. 1996. *Lernpsychologie*. Psychologie Verlagsunion, Weinheim, 5 edition.

E. Hagen, T. Said, and J. Eckert. 2004. Spracheingabe im neuen BMW 6er. *Sonderheft ATZ/MTZ (Der neue BMW 6er)*, May:134–139.

R. Haller. 2003. The Display and Control Concept iDrive - Quick Access to All Driving and Comfort Functions. *ATZ/MTZ Extra (The New BMW 5-Series)*, August:51–53.

L. Hassel and E. Hagen. 2005. Evaluation of a Dialogue System in an Automotive Environment. In *6th SIGdial Workshop on Discourse and Dialogue, Lisbon, Portugal, 2-3 September 2005*. Draft Version.

P. Heisterkamp. 2001. Linguatronic - Product-Level Speech System for Mercedes-Benz Cars. In *Proceedings of the 1st International Conference on Human Language Technology Research (HLT-01), San Diego, CA, USA*.

T. K. Landauer. 1997. Behavioral Research Methods in Human-Computer Interaction. In Martin A. Helander, Thomas K. Landauer, and Prassad V. Prabhu, editors, *Handbook of Human-Computer Interaction - second, completely revised edition*, pages 203–227. North-Holland, Amsterdam, Lausanne, New York u.a.

L. Libuda. 2001. Improving Clarification Dialogs in Speech Command Systems with the Help of User Modeling: A Conceptualization for an In-Car User Interface. In *Online-Proceedings des 9. GI-Workshops: ABIS-Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*. GI-Fachgruppe: Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen (ABIS).

R. R. Mourant, F.-J. Tsai, T. Al-Shihabi, and B. K. Jaeger. 2001. Divided Attention Ability of Young and Older Drivers. Proceedings of the 80th Annual Meeting of the Transportation Research Board.

J. Nielsen. 1993. *Usability Engineering*. Academic Press Professional, Boston u. a.

W. Piechulla, C. Mayserb, H. Gehrke, and W. König. 2003. Reducing Drivers' Mental Workload by Means of an Adaptive Man-Machine Interface. *Transportation Research Part F: Traffic Psychology and Behaviour*, 6(4):233–248.

E. Rich. 1979. User Modeling via Stereotypes. *Cognitive Science*, 3:329–354.

S. Rogers, C.-N. Fiechter, and C. Thompson. 2000. Adaptive User Interfaces for Automotive Environments. In *Proceedings of the IEEE Intelligent Vehicles (IV) Symposium 2000, Detroit (USA)*, pages 662–667.

Jing Wu. 2000. Accomodating both Experts and Novices in One Interface. *Universal Usability Guide*. Department of Computer Science, University of Maryland, http://www.otal.umd.edu/UUGuide/.