Evaluation of a Dialogue System in an Automotive Environment

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

In this paper we discuss features to enhance the usability of a spoken dialogue system (SDS) in an automotive environment. We describe the tests that were performed to evaluate those features, and the methods used to assess the test results. One of these methods is a modification of PARADISE, a framework for evaluating the performance of SDSs (Walker et al., 1998). We discuss its drawbacks for the evaluation of SDSs like ours, the modifications we have carried out, and the test results.

1 Introduction

SDSs for operating devices are still a small group in the class of dialogue systems. But, as available applications and research in this area show, there is a growing demand for such interfaces. They are being used in the mechanical CAD industry (think3, 2000) and in smart home systems (Fellbaum and Hampicke, 2002). The operation of complicated devices by voice, e.g. programming a video-tape recorder, was surveyed by Aretoulaki and Ludwig (1998). Our domain is the operation of devices like radio, navigation, and telephone while driving a car (Haller (2003); Hagen et al. (2004)).

We proposed features to enhance the usability of the system. These improvements are based on guidelines for SDSs in general, e.g. on the work of Dix et al. (1995), and on principles for in-car SDSs, e.g. Dybkjær et al. (1997) and the guideline ETSI EG 202 116 V 1.2.1 (2002, section 8.7.3). These features were implemented in a prototype and surveyed in two test series, one with a reference system (Hagen et al., 2004) and the other with the prototype. The importance of testing in a real environment was pointed out by Bernsen and Dybkjær (2001). Thus evaluation of the prototype took place driving in real traffic.

A frequently mentioned framework to evaluate SDSs is PARADISE (Walker et al., 1998). It seeks to predict system performance (described in terms of the user satisfaction) employing multiple regression analysis using a task success metric based on the Kappa value, κ (Carletta, 1996), and dialogue costs as independent variables. We revised κ for our system because it was developed on the basis of information dialogues with a well defined set of task attributes, what is not the case for the dialogues we evaluated. Despite this modification, we found no relationship between task success and dialogue costs, and user satisfaction. We discuss the reasons for this issue.

In section 2 we describe the SDS, and section 3 explains the features. In section 4 and 5 we describe the test design and the methods used to evaluate the tests. In sections 6 and 7 we present the findings of the evaluations. Section 8 summarizes these results.

2 System Description

Our speech interfaces were 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, fig. 1) and central display in the centre column (fig. 2). When users operate the controller (turn left and right, push in four directions and press down), they receive visual feedback on the display.

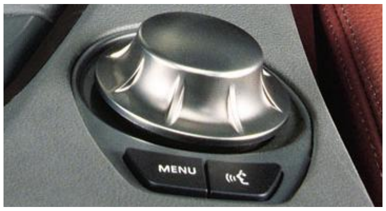


Figure 1: Controller and PTT-button

Over the speech channel, users can operate functions in the areas entertainment, communication and navigation. Users activate the speech recognizer with a push-to-talk (PTT) button on the steering wheel or in the middle console near the controller. The dialogue style is command

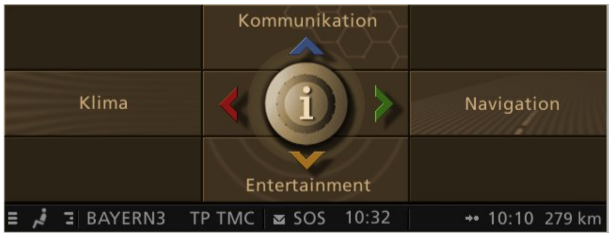


Figure 2: Display Control

and control as illustrated in table 1. The iDrive SDS is currently configured for about 3000 words and phrases. iDrive with speech is available in several languages. For our experiments, we used the German version. For further information, see Hagen et al. (2004).

3 Features for Enhancing Usability

Usability is a multidimensional property of a user interface. The definition we use is based on Nielsen (1993). There, five usability dimensions are mentioned: Learnability, Efficiency, Memorability, Error, and Satisfaction. According to Nielsen (1993), a system fulfills the demands of usability when it is easy to learn (U-1), efficient to use (after the learning phase, U-2), easy to be remembered (U-3), when it allows an easy recovery from errors (U-4), and it is pleasant ot use (U-5).

We aim at enhancing the usability of the system. The features discussed below help the SDS to conform to these requirements. We have classified the features according to the degree of control users have over them in *implicit* (**I**, section 3.1) and *explicit* (**E**, section 3.2).

3.1 Implicit Features

With the help of the implicit features, the system adapts to the users' behavior (I-1, I-2, and I-3) and provides means to facilitate its use (I-2 and I-4).

I-1: The system prompts are adapted to the expertise of the users. For novices, the SDS mentions the available voice commands (options) without waiting for users to ask. Experts have to explicitly ask for options (table 1). This feature is part of the adaptation concept described in (Hassel and Hagen, 2005). Feature I-1 makes the system easy to learn (U-1). It also improves the interaction efficiency once users have learned how to use it (U-2), because the reduced prompts save time. And it makes the system more pleasant to use (U-5), because novices, due to the informative prompts, do not feel lost, and experts are not annoyed by long and repetitive prompts.

I-2: Certain tasks are more efficiently executed with a voice command than with the controller and GUI. In such cases, the system takes the initiative and suggests to switch modality. I-2 improves the learnability of the iDrive (U-1) because it tells users which modality is the more appropriate to complete the current task, with controller or by voice. Feature I-2 was only available in the prototype. The experiments we carried out were restricted to the SDS. Therefore, we could not test this multimodal feature.

I-3: Timeouts and ASR-failures cannot be completely avoided. Timeouts occur in most cases because drivers are distracted by the traffic environment or because they do not know what to say next. After the first timeout, the system repeats the prompt to catch the attention of the driver. After the second successive timeout, the system prompts the currently available options. Due to the limited limited vocabulary, ASR-failures because of OOV mistakes can happen. If the system does not understand users after two tries the system prompt is changed to contain the currently available options. Feature I-3 makes it easy for users to learn the system (U-1) and to recover from errors (U-4).

I-4: The *Speak What You See* principle means that users are able to use the words or phrases labelling tasks appearing on the GUI as voice commands. This principle diminish the users' need for remembering the commands (U-3) because they can look at the GUI to recall the available voice commands.

3.2 Explicit Features

With the help of the explicit features, users can actively control what and when they learn. These features guarantee that users keep control over the system, they are in charge of the information they get from the SDSs.

E-1: With the "help" command users can learn about the general characteristics of the system: how to get a list with voice commands, how to get to the main menu, etc. It facilitates users to understand the system (U-1). After "options" (E-2) the system prompts the currently available voice commands. The effect of this command is context sensitive. Feature E-2 facilitates novices to learn the system (U-1) and experts to learn about tasks they seldom use (U-2). Both E-1 and E-2 makes the system more pleasant to use (U-5) because drivers do not need to look in the printed manuals for advice.

E-3: Users can ask the system to suggest them a faster way to achieve the actual task. The system looks for shortcuts to achieve one of the last dialogue states and suggests it to the user. Feature E-3 allows users to learn more efficient ways to use the SDS (U-2).

E-4: The "back" command has a similar effect as the *back button* of a browser. During the first test series (reference system) some users tried to recover from misunderstandings using the command "back". Users expecting the command to be available were astonished and confused about its absence. E-4 allows users an easy recovery from errors (U-4), thus facilitating the learning by

Novice		**Expert**	
user:	<presses PTT button>	user:	<presses PTT button>
system:	Speech input <beep>	system:	<beep>
user:	Entertainment.	(user:	Entertainment.)
system:	Entertainment. Say ’FM menu’, ’AM menu’, or ’CD menu’.	(system:	Entertainment.)
user:	FM menu.	(user:	FM menu.)
system:	FM menu. Say ’choose frequency’, ’choose station’, ...	(system:	FM.)
user:	Choose frequency.	user:	Choose frequency.
system:	Which frequency do you want?	system:	Enter frequency.
user:	96.3	user:	96.3
system:	You are hearing 96.3 MHz.	system:	<music is heard>

Table 1: Sample Dialogue

trial and error (U-1) as well as, indirectly, enhancing the efficiency of the system usage (U-2).

E-5: The ”up” command allows users to navigate upwards in the GUI. Other than ”back”, ”up” does not undo user instructions. It only moves the focus from one layer to the one above. Using it, users can recover from misunderstandings (U-4), and abbreviate the interaction (U-2).

The impact of each feature on usability is presented and dicussed in section 7. The evaluation showed that the proposed features do contribute to enhance the usability of the SDS.

4 Test Design

The prototype described in section 2 was evaluated against a reference system with the same functionality and the same GUI (Hagen et al., 2004). Two test series were carried out. For series A, a BMW 5 Series was equipped with a reference system. Series B with the prototype took place in a BMW 7 Series. A total of 44 subjects participated in the tests. Table 2 summarizes the participants’ characteristics.

Test Series	A (Reference System)	B (Prototype)
Mean Age **(Range)**	28,77 (21 - 43 years old)	25,64 (22 - 33 years old)
Number **of Subjects**	22 (15 male, 7 female)	22 (15 male, 7 female)

Table 2: Comparison of the Test Series

The tests consisted of two parts, a driving part (duration: between 30 and 45 min) and a questionnaire. During the driving part the subjects were asked to complete eleven representative tasks (table 3). Tasks 1 and 2 were repeated at the end of the test (tasks 10 and 11) to test the adaptation of the system and the learning progress of the participants: Could they achieve the task more efficiently? Did they already develop an operating strategy during the test time?

In addition to completing the tasks while driving, users were told to verbalise their thoughts as they used the system. The thinking-aloud method is described by Nielsen (1993). After finishing the driving part, the test participants had to answer a five-page questionnaire.

Task 1:	choose frequency 93.3
Task 2:	choose station bayern 5
Task 3:	play title number 4 of the current cd
Task 4:	activate traffic programm
Task 5:	dial a phone number
Task 6:	dial a name from the address book
Task 7:	display the navigation map
Task 8:	change the map scale to 100 m
Task 9:	change the map style (north, driving, arrows)
Task 10:	choose an arbitrary frequency
Task 11:	choose an arbitrary station

Table 3: Test Tasks

5 Evaluation Method

To assess the test results we intended to use the evaluation framework PARADISE (Walker et al., 1998). In the last years, PARADISE was often surveyed (Whittaker et al. (2000); Paek (2001); Larsen (2003b); Aguilera and et al. (2004)). The main limitation was found to be that tasks have to be clearly defined so that they can be described by an attribute-value-matrix (AVM). Further, it was critized that PARADISE was designed to evaluate only unimodal systems. And lastly, the assumption of a linear relationship between user satisfaction and subjective measures was called into question.

Attempts have been made to revise PARADISE. Hjalmarsson (2002) propose a new task definition for the evaluation of multimodal systems with non-AVM-describable tasks. We could not apply this method because they evaluated SDSs for information exchange and the task success was calculated in terms of information bits. Beringer et al. (2002) also introduce a new task success measure to evaluate multimodal systems. They rate tasks as successful or not, but since we wanted to know *how well* users coped with the tasks, we also discarded this method. In the next sections we describe the changes we carried out to PARADISE in order to apply it to our system.

5.1 A Modified κ Calculation

SDSs for the car environment offer users a broad spectrum of tasks, e.g. dialing a telephone number, setting navigation options and tuning a radio frequency. The type of tasks in this environment can be represented by a directed, connected graph with marked and unmarked nodes (fig. 3), through which users navigate and where the task is completed after they reach the desired node. The edges represent the transitions due to user utterances, and the nodes represent states of that dialogue space. Only a few edges were drawn, subdialogues (options and help requests, etc.) as well as the transitions caused by the command "back" were left out. Marked nodes are drawn with heavy line, and utterances are set in quotation marks. Unmarked nodes are transitional states: the SDS remains active after users have reached such states, and the dialogue strategy remains user initiated.

Fragment A in figure 3 presents two possibilities: Users can navigate to the node **View** by choosing a view in the navigation menu (north, driving, arrows) - in figure 3 users chose "arrow view", or they can navigate to the node **Scale** by saying they want to change the scale of the map. In this last case, the system takes the initiative asking users what scale they want to have (table 4). In B users navigate to the node **Dial Number**, where they are asked to enter a telephone number. This subdialogue is displayed inside the node.

user:	Navigation menu.
system:	Navigation. You can say route criteria, map, ...
user:	Map.
system:	Map. You can say map style, or change scale.
user:	Change scale.
system:	Choose a scale.
user:	200 meters
system:	Scale changed to 200 meters

Table 4: Dialogue Leading to the **Scale** Node in Figure 3

When users reach a marked node, usually either the dialogue is done immediately (node **View**), or the system takes the initiative to require information from the users, and then the dialogue is done (nodes **Scale** and **Dial Number**). But whether a task has been completed or not is not always that easy to answer. The crux of the matter is the goal of the users: If they just want to have the phone menu displayed, then the task is done after they reach the node **Phone** (fig. 3). In our SDS, tasks cannot be described in terms of AVMes.

Since our dialogues can not be represented by AVMs we had to define κ in a different way. Instead of task attributes, we have specified for each task a set of nodes starting from the main menu and following the usual paths to the nodes that represent the test tasks. Figure 4 shows the AVM of the task 5 (dial a phone number), represented as a graph in figure 3 part B. Since the tasks for the tests are fixed, for each task a subset of nodes defines when it is complete. The black diagonal cells **Ready** represent the final states.

In PARADISE only utterances referring to task attributes are recorded in the AVM. We also include those that contribute indirectly to accomplishing the tasks. For this purpose we introduce the following attributes: OPTIONS/HELP, STOP, REPEAT, FAILURE, and BACK (for the prototype). FAILURE subsumes answer failures due to a voice recognition misunderstanding (grey columns in figure 4), answer failures due to a wrong user input (last diagonal cells) and correct system answers due to wrong user utterances (grey rows).

PARADISE computes only correctly recognised utterances or "misunderstandings that are not corrected in the dialogue" because "the effect of misunderstandings that are corrected during the course of the dialogue are reflected in the costs associated with the dialogue" (Walker et al., 1998). Such an AVM is supposed to "summarize how well an agent achieves the information requirements of a particular task" (Walker et al., 1998). But, since our dialogues are not based on information requirements, we do not have a set of attributes that have to be accomplished for the task to be successful. Therefore, we consider all utterances that occur during the dialogue in order to compute κ. Following (Walker et al., 1998), we consider the FAILURE cells in the calculation of the total number of utterances, but exclude it from the calculation of $P(A)$ and $P(E)$. Such an AVM summarizes how well users coped with the task.

κ is usually used to measure pairwise agreement among a set of coders making category judgments, correcting for chance expected agreement (Siegel and Castellan, 1988). There, $P(A)$ is the proportion of times that the coders agree and P(E) is the proportion of times that one would expect them to agree by chance (κ formula 1). This Kappa, we called it κ^*, is calculated in a slightly different way than in PARADISE (κ^P). The definition of $P(A)$ is the same in both cases (formula 2[1]). In PARADISE, $P(E)$ is calculated using only the columns of the matrix (formula 3), thus taking only the exchanged information into consideration, independently from who uttered it, system or users. The standard calculation of $P(E)$ includes rows and columns (formula 4), so that using κ^* both system's and user's side are taken into con-

[1] In this definition $p_{.x}$ is equivalent to $\frac{t_x}{T}$ in PARADISE.

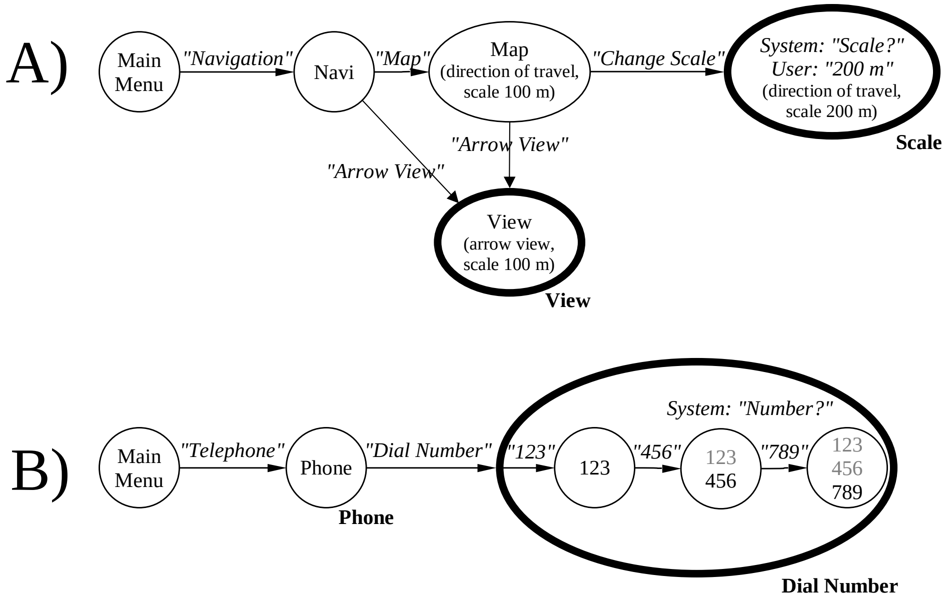
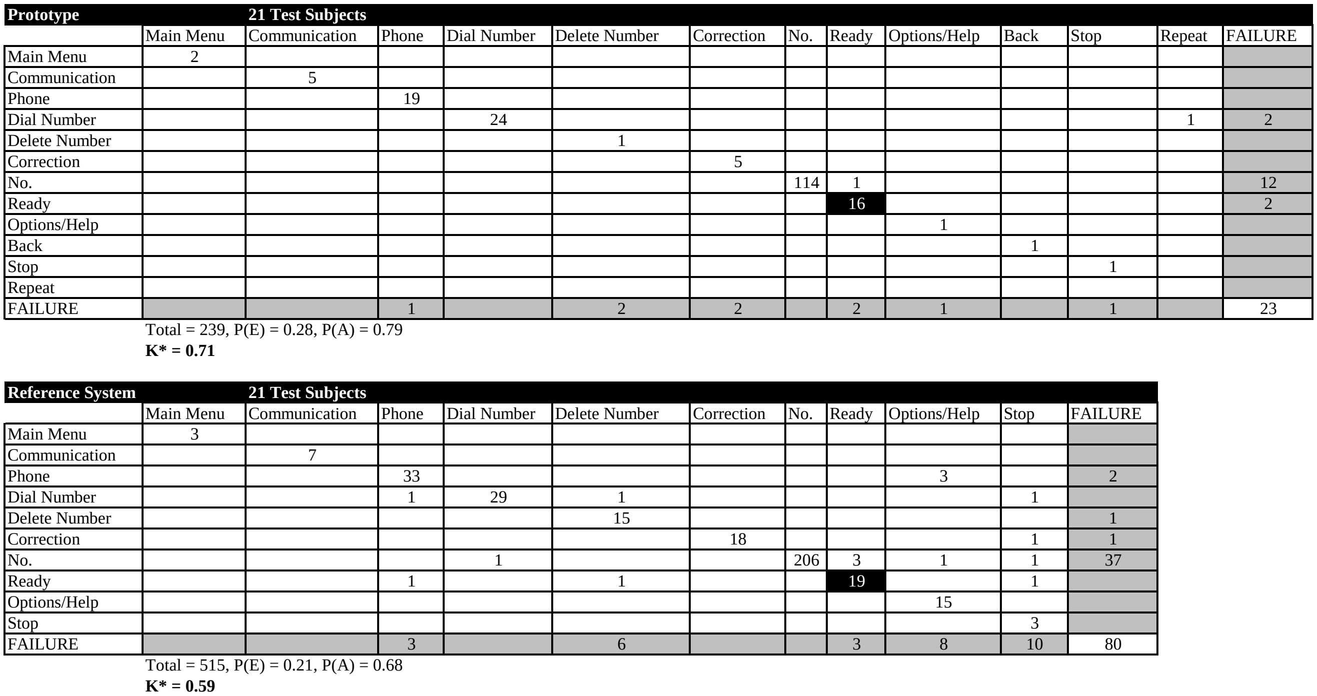


Figure 3: Fragments of the Dialogue Space



Prototype		**21 Test Subjects**											
	Main Menu	Communication	Phone	Dial Number	Delete Number	Correction	No.	Ready	Options/Help	Back	Stop	Repeat	FAILURE
Main Menu	2												
Communication		5											
Phone			19										
Dial Number				24								1	2
Delete Number					1								
Correction						5							
No.							114	1					12
Ready								16					2
Options/Help									1				
Back										1			
Stop											1		
Repeat													
FAILURE			1		2	2		2	1		1		23

Total = 239, P(E) = 0.28, P(A) = 0.79
K* = 0.71

Reference System		**21 Test Subjects**									
	Main Menu	Communication	Phone	Dial Number	Delete Number	Correction	No.	Ready	Options/Help	Stop	FAILURE
Main Menu	3										
Communication		7									
Phone			33						3		2
Dial Number			1	29	1					1	
Delete Number					15						1
Correction						18				1	1
No.				1			206	3	1	1	37
Ready			1		1			19		1	
Options/Help									15		
Stop										3	
FAILURE			3		6			3	8	10	80

Total = 515, P(E) = 0.21, P(A) = 0.68
K* = 0.59

Figure 4: Calculation of κ^* for Task 5 (Dial a Phone Number)

sideration. We have calculated κ^* and κ^P to see which one correlates better with our data.

$$\kappa = \frac{P(A)-P(E)}{1-P(E)} \qquad (1)$$

$$P(A) = \quad p_{11} + p_{22} + p_{33} + \cdots + p_{nn} \qquad (2)$$

$$P(E) = \quad p_{.1}^2 + p_{.2}^2 + p_{.3}^2 + \cdots + p_{.n}^2 \qquad (3)$$

$$P(E) = \quad p_{1.}p_{.1} + p_{2.}p_{.2} + p_{3.}p_{.3} + \cdots + p_{n.}p_{.n} \qquad (4)$$

For a better understanding of the formulas listed above, we display a matrix to illustrate the meaning of the used terms. A to N are the attributes, p_{xy} are the number of times an attribute was chosen divided by the total number of utterances, T, and $p_{x.}$ and $p_{.x}$ are the sum of all values in row x over all columns and the sum of all values in column x over all rows, respectively:

$$\begin{array}{c|cccc|c}
 & A & B & \cdots & N & \\
\hline
A & p_{11} & p_{12} & \cdots & p_{1n} & p_{1.} \\
B & p_{21} & p_{22} & \cdots & p_{2n} & p_{2.} \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
N & p_{n1} & p_{n2} & \cdots & p_{nn} & p_{n.} \\
\hline
 & p_{.1} & p_{.2} & \cdots & p_{.n} &
\end{array}$$

The data analysed so far from our experiments did not confirm the claim of a correlation between user satisfaction and κ^* together with the cost factors. Beside κ^*, we used the cost factors barge-in, help and option requests, and number of turns (section 6) as independent variables to calculate the performance function. Before the calculation all values were normalized to z-scores, so that we could easily control if there were outliers that would have distorted the comparison, but this was not the case. Using $US1$ (section 7) as dependent variable, we obtained for the system B a coefficient of determination $r^2 = 0.07$. Therefore, we can not apply the multivariate linear regression proposed in PARADISE to calculate a performance function for our systems. In spite of that, we found κ^* to be a good measure to characterize how difficult it was for users to accomplish (or try to accomplish) the task. Further analysis of the data will show if this assumption is right.

6 Evaluation of the Driving Part

In this section we present the results of our test series. For our evaluation we use the usual metrics as described in Larsen (2003a) and NIST (2001). We compare the following cost factors for systems A and B: Task duration, number of turns, task success, number of barge-in attempts at the beginning of system utterances, and number of option and help requests.

6.1 Task Duration

Figures 5 shows how long it took the users to complete the different tasks in the two systems. Test subjects for series A (TS_A) needed on average 62.1 sec to complete a task, and test subjects for series B (TS_B) 47.0 sec[2]. Seven of the eleven tasks were accomplished faster with system B than with A. The results for the other four tasks differ from what was expected: First, the longer task completion times for tasks 7 and 9 in the prototype. This can be largely explained by the circumstance that test subjects were all novices. The system prompts of B for these tasks were much longer than the ones of A, for example:

user: Navigation.
system: Navigation, you can say new destination, last destinations, route, change map style, or change map scale.

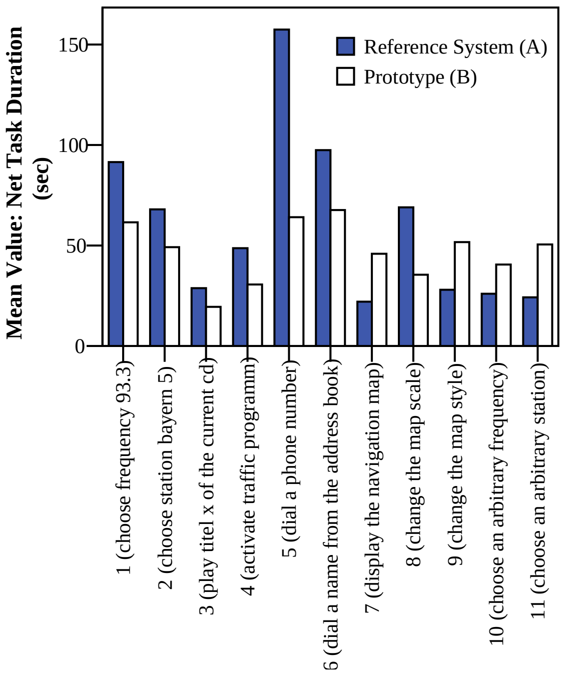


Figure 5: Task Duration

Second, the different progress between tasks 1 and 2, and their reiterations, 10 and 11, for both systems. The task duration for task 10 is in both systems lower than in task 1, but more remarkable in system A. Users of this system needed for task 10 on average only 30% of the time they needed for task 1. Task duration of task 11 decreased only in system A. In system B these values remained almost the same as for task 2. This may indicate that users of the reference system learned faster that they can speak the tasks they want to activate directly (shortcuts). The help given to the novices in the prototype seems to slow down this insight among the users of this system. They repeatedly applied the same tactics, they followed the menu structure of the system instead of speaking the desired commands directly. The effect on user satisfaction will be discussed in section 7.

6.2 Number of Turns

Figure 6 shows how many turns users needed to complete the different tasks in the two systems. TS_A needed on average 8.7 turns to complete a task, and TS_B 6.9 turns. Seven of the eleven tasks were accomplished with less interactions in system B than in A. The results for the other four tasks differ from what was expected. First, TS_B needed more turns to complete tasks 7 and 9 than

[2] Interruptions due to traffic conditions were documented during the test and then used to adequately rectify the times.

TS_B. This can be explained by the kind of system utterances TS_B got. Test subjects were all novices and, therefore, these utterances told the users which commands they could speak next. Most users employed exactly the commands offered by the system, what lead them to follow each time the menu structure rather than skipping nodes, i.e. using shortcuts.

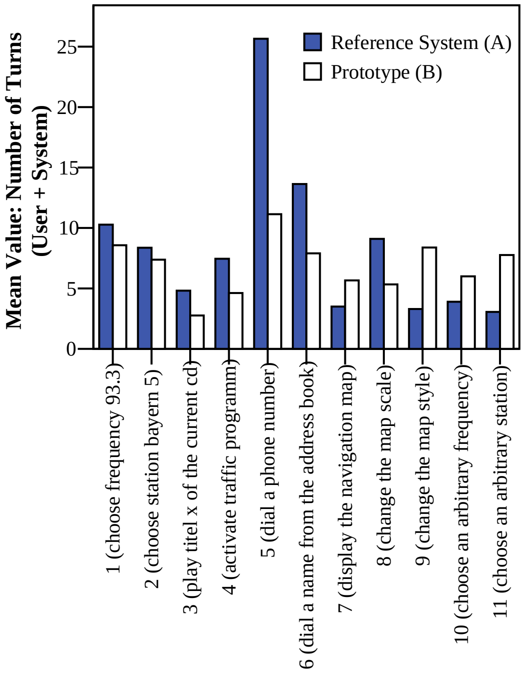


Figure 6: Number of Turns

Second, the different progress between the tasks 1 and 2, and their reiterations, 10 and 11, for both systems. TS_A needed for task 10 only ca. 30% of the turns they needed for task 1. TS_B still needed ca. 66% of the turns to complete the task. The number of turns of task 11 decreased only in system A. For system B these values remained almost the same as for task 2. These can also be explained by the kind of utterances in system B. In many cases, these users were treated as experts while solving tasks 1 an 2 the second time, i.e. tasks 10 and 11. However, they behaved as they had learned and skipped almost no nodes.

As tables 5 and 6 show, there exists a high correlation between task duration and number of turns. Therefore, either can be used for calculating the performance function. 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. Comparing tasks 1 and 2 with 10 and 11 (tables 5 and 6), we observed that TS_B reached very fast the asymptote of the curve, i.e. users learned very fast how to use our prototype. The system prompts for novices served their purpose. Our tests confirmed that the initial part of the learning curve for the prototype's users corresponds to the recommended shape.

TS_A learned by trial and error that they can speak the tasks they want to activate directly, leaving out the nodes between. The first time they completed tasks 1 and 2 they were not so successful as TS_B, but they were more efficient the second time they completed those tasks. According to Nielsen (1993) systems designed for experts are hard to learn but highly efficient, i.e. the learning curve is even at the beginning (Nielsen, 1993). 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 to learn better ways of completing a task (cf. section 3.2). Long term experiments still have to show if system B displays a typical expert learning curve over the time.

6.3 Task Success

Figure 7 compares the task success rates for both systems. For system B the mean success rate reached 94%, system A's mean success rate was 78%. Only 3% of the tasks could not be completed at all, in either system, usually because users gave up. Ca. 15% of the tasks in system A, and ca. 3% of the tasks in system B were accomplished only partly, most frequently because users were a bit confuse and asked the experimenter for a hint or because they said the right command but the system did not understand. The ASR system was the same in both series, therefore, the main reason for this difference (15% and 3% for system A and B) was that TS_B were less confuse about what to say next. This confirms the benefit of telling novices the available commands.

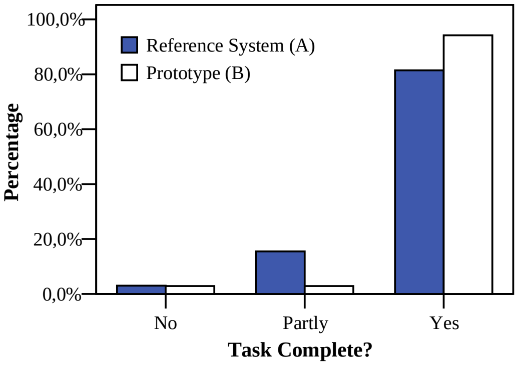


Figure 7: Did Users Complete the Tasks?

6.4 Other Cost Factors

Figure 8 shows a comparison of four cost factors (number of option, help, and OOV-help requests, and number of barge-in attempts at the end of system prompts). The commands ”help” and ”options” are described in secion 3.2. Under OOV-help requests we added up help requests for which users employed OOV-words. With the cost factor barge-in we consider utterances spoken before the ASR was listen. All four factors were divided by the number of turns needed to accomplish each task. The comparison of relative values allows us to subtract the influence of number of turns from these cost factors.

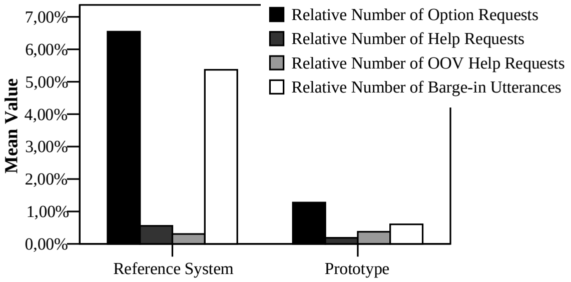


Figure 8: Relative Number of Option Requests, Help Requests, and Barge-in.

The dialogue costs were in every task lower for the prototype than for the reference system. Most remarkable is the decrease of option requests for system B. TS_B asked only a $\frac{1}{5}$ of the times TS_A did. The reason was that TS_B got the available commands from the system, without having to ask for them. Therefore, they usually knew what to say. The same applies to the number of help requests. TS_B asked for help $\frac{1}{3}$ of the times TS_A did. The number of OOV help requests was for both systems almost the same. Barge-in was in series B nine times more frequent than in series A. This system signaled users that they could speak with a tone at the end of every prompt, only then the ASR was active. System B relied on the turn taking theory of the conversational analysis (Clark, 1997) and omitted that additional auditive turn taking signal. The strong decrease of commands uttered ahead of time verified that this strategy was the more natural. The comparison of these four cost factors confirms that users cope better with the prototype.

7 Evaluation of the Questionnaire

The questionnaire uses a Likert scale with four choices ranging from strongly opposed (1) to strongly in favour (4). It consists of four parts: questions about the participant, about his technical background, about the test (users’ attitude towards the system), and about the system (how users judge the system’s ergonomics).

We calculated two factors to measure the user satisfaction (US$_1$ and US$_2$). US$_1$ subsumes three answers to questions about the test: ”I could complete all tasks without problems”, ”I find the system easy to use”, and ”I got frequently upset during the test”. US$_2$ subsumes three answers to questions about the system: ”I would recommend the system”, ”I really want to have such a system in my car”, and ”I find the system very useful”. Figures 9 and 10 show the values for US$_1$ and US$_2$ for both systems and over the task completion rate. Users rated US$_1$ and US$_2$ better for series B than for series A.

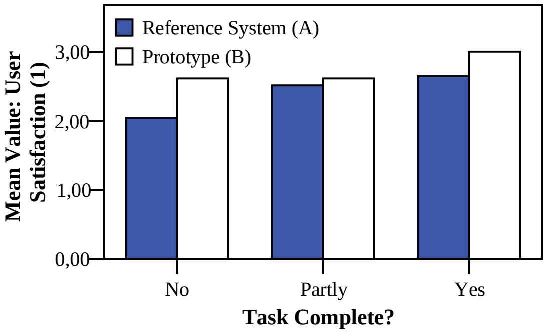


Figure 9: User Satisfaction (1)

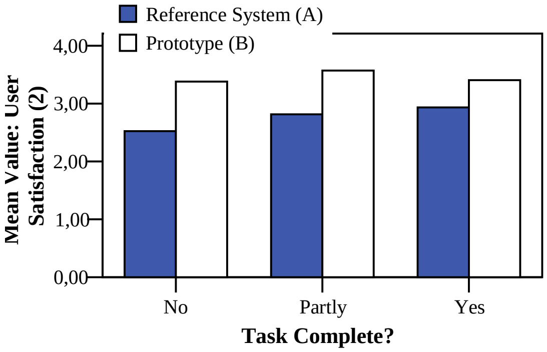


Figure 10: User Satisfaction (2)

We summarize users’ evaluation of the features in table 5. The features marked with an asterisk were provided only in system B. The other features were available in both systems. In the second column we have specified the effect on usability of each feature. Below we discuss the evidence found for each feature (I-1 to I-4, and E-1 to E-5). The statements in brackets are questions from the questionnaire, and the percentages allude to the number of users (strongly) agreeing with the statement.

I-1: Many more TS_B than TS_A (almost) never looked at the GUI. They did not need to do so because the knew

Feature	Effect on Usability
I-1* (Information content of prompts)	Positive ((Almost) never looked at the display: $TS_B : 68\%, TS_A : 45\%$. Options should be prompted every time: $TS_B : 77\%, TS_A : 27\%$)
I-2* (System initiative)	Not specified
I-3 (Timeouts, ASR-Failures)	Positive (Help prompts were not (at all) helpful: $TS_B : 36\%, TS_A : 64\%$)
I-4 (Speak what you see)	Positive (Completely lost without the display: TS_B : $36\%, TS_A : 68\%$)
E-1 (Help)	Contradictory (Didn't know how to ask for help: TS_B : $50\%, TS_A : 64\%$)
E-2 (Options)	Contradictory (Didn't know the difference between help and options: $TS_B : 73\%, TS_A : 50\%$)
E-3* (Suggestion)	Positive (Desirable feature: TS_B : 82%, TS_A : 68%)
E-4* (Back)	Positive (Command is absolutely necessary: TS_B : $100\%, TS_A : 86\%$ It was easy to rectify a misunderstanding: $TS_B : 45\%, TS_A : 27\%$)
E-5* (Up)	Neutral

Table 5: Features and Test Results
(*feature is provided only in the prototype)

the commands. In general, users found the enumeration of the available options a good means to learn the system. Therefore and because they knew they could asked for options and help (E-1, E-2), they approved of adaptation.

I-3: Considerable more TS_A than TS_B asserted that ”help prompts were not (at all) helpful” did. This difference may be explained by the time help was issued in both systems. While in series B novices got help right away after saying a command, TS_A had to wait the second ASR-failure or timeout to get system initiated help. At that time, many users were already confused and found the offered options not so helpful anymore.

I-4: The tests also confirmed the importance of the graphical context for usability. Users expect the text on the GUI to be voice commands.

E-1, E-2: The results about these features were contradictory. On the one hand, more TS_A than TS_B stated that they did not know how to ask for help. But, on the other hand, every test subject asked at least once for help, using either the ”help” or the ”options” command. Maybe they were not aware of it, but they use the commands in an instinctive way.

E-3: The ”suggestion” command was rated differently by TS_A and TS_B. While TS_A had some doubts about this feature, TS_B, having tested it, approved of it.

E-4: The tests verified that error recovering is normally very difficult to deal with for users, and that users' expectations due to knowledge transfer are extremely persistent (Norman, 2002). Therefore, the ”back” command had broad acceptance among users.

E-5: The command ”up” had not the same positive impact on the usability of the system as ”back”. Thus, the contribution for the usability improvement of this command does not justify the expensive implementation.

8 Conclusion

We calculated two task success measures based on PARADISE, κ^P and κ^*, but we could not find a linear relation between US and task success plus cost factors. Consequently, we could not use these methods to calculate system performance. However, κ^* proved to be appropriate to assess how difficult it was for the users to accomplish (or try to accomplish) the task. Table 6 shows a comparison of κ^* values for tasks 1 to 5 for both systems. These values show that users dealt better with the prototype.

Task	Series A	Series B
1	.33	.54
2	.33	.47
3	.55	.80
4	.44	.57
5	.59	.71

Table 6: κ^* for Reference System (A) and Prototype (B)

Users' levels of satisfaction US_1 and US_2 were almost completely unrelated to success rates. One reason for this finding may lie in the novelty of voice interfaces in the automotive environment. The characteristics of the test subjects largely agreed with those of early adopters: young, urban, and highly educated. For such users, the main goal of operating an innovative system is the interaction itself, not task completion. Experiments with real customers should be carried out to confirm this hypothesis.

Another reason for the absence of correlation might be the redundancy of the system. Voice interface is not the only input device but an additional possibility, besides the manual input, to operate the comfort tasks at disposition in the car. Therefore, the requirements of the users are others than, e.g. for telephony SDSs, where the voice interface is the sole input device.

All subjective and nearly all objective measures were better for series B. Test persons had not used the voice interface in the car before. The results of our evaluations confirm the expected positive effects of prompt adaptation and the other proposed features. But we do not know how experts would cope with the systems. On the one hand, the comparison of tasks 1 and 2 with their repetitions 10 and 11 showed that the learning curve was very steep for system B. On the other hand, tasks 7 and 9 suggest that the extended prompts for novices in system B could lead users to operate the system in a less straightforward manner than system A because they did not use shortcuts. The prompts of system B become the same as in system A when users turn experts. Will experts change their habits and learn the shortcuts? Long term evaluations have to be performed to investigate the benefit of the proposed features over time.

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

E. J. Gómez Aguilera and N. O. Bernsen et al. 2004. Usability Evaluation Issues in Natural Interactive and Multimodal Systems - State of the Art and Current Practice (Draft Version). Technical report. Project SIMILAR SIG7 on Usability and Evaluation, Deliverable D16.

M. Aretoulaki and B. Ludwig. 1998. Skizzierung eines allgemeinen Szenarios für Bediendialoge . Jahresbericht 1998 der Forschungsgruppe Wissensverarbeitung am Bayerischen Forschungszentrum für wissensbasierte Systeme, http://www-wv.informatik.uni-erlangen.de/fg-wv/.

N. Beringer, U. Kartal, K. Louka, F. Schiel, and U. Türk. 2002. PROMISE - A Procedure for Multimodal Interactive System Evaluation. Technical report, LMU München, Institut für Phonetik und sprachliche Kommunikation. Teilprojekt 1: Modalitätsspezifische Analysatoren, Report Nr. 23.

N. O. Bernsen and L. Dybkjær. 2001. Exploring Natural Interaction in the Car. In *International Workshop on Information Presentation and Natural Multimodal Dialogue*, pages 75–79, Verona, Italy 14-15 Dec. 2001.

J. Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

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

A. Dix, J. Finlay, and G. Abowd. 1995. *Mensch Maschine Methodik*. Prentice Hall.

L. Dybkjær, N. O. Bernsen, and H. Dybkjær, 1997. *Designing Co-Operativity in Spoken Human-Machine Dialogue*, volume 2 of *Research Reports Esprit*, pages 104–124. Springer Verlag.

ETSI EG 202 116 V 1.2.1, 2002. *Human Factors (HF); Guidelines for ICT Products and Services; Design for All*. European Telecommunications Standards Institute (ETSI).

K. Fellbaum and M. Hampicke. 2002. Human-Computer Interaction in a Smart Home Environment. In *4th International Congress on Gerontechnology, Miami Beach, USA*, pages 1–6, November 9–12.

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. Adaptation of an Automotive Dialogue System to Users's Expertise. In *6th SIGdial Workshop on Discourse and Dialogue, Lisbon, Portugal, 2-3 September 2005*. Draft Version.

A. Hjalmarsson. 2002. Evaluating AdApt, a Multi-Modal Conversational, Dialogue System Using PARADISE. Master's thesis, Department of Speech Music and Hearing, KTH Royal Institute of Technology, Stockholm.

L. B. Larsen. 2003a. Evaluation Methodologies for Spoken and Multi Modal Dialogue Systems - Revision 2. May 2003 (Draft Version). Presented at the COST 278 MC-Meeting in Stockholm 2.-4. May 2003.

L. B. Larsen. 2003b. Issues in the Evaluation of Spoken Dialogue Systems using Objective and Subjective Measures. In *Proceedings of the 8th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, St. Thomas, U.S. Virgin Islands, Dec. 2003.

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

NIST. 2001. Common Industry Format for Usability Test Reports - Version 2.0, May 18, 2001. Technical report, National Institute of Standards and Technology.

Donald A. Norman. 2002. *The Design of Everyday Things*. Basic Books, New York.

T. Paek. 2001. Empirical Methods for Evaluating Dialog Systems. In *ACL 2001 Workshop on Evaluation Methodologies for Language and Dialogue Systems, Toulouse, France*.

S. Siegel and N. J. Castellan. 1988. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill International, Singapore.

think3. 2000. Think3: thinkdesign 6.0 Debuts To Rave Reviews. Press Releases, http://www.think3.com/en/news/.

M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. 1998. Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies. *Computer Speech and Language*, 12(3):317–347.

S. Whittaker, L. Terveen, and B. A. Nardi. 2000. Let's Stop Pushing the Envelope and Start Addressing It: A Reference Task Agenda for HCI. *Human Computer Interaction*, 15:75–106.