Adaptive Help for Speech Dialogue Systems Based on Learning and Forgetting of Speech Commands

Alexander Hof, Eli Hagen and Alexander Huber
Forschungs- und Innovationszentrum
BMW Group, Munich
alexander.hof,eli.hagen,alexander.hc.huber@bmw.de

Abstract

In this paper we deal with learning and forgetting of speech commands in speech dialogue systems. We discuss two mathematical models for learning and four models for forgetting. Furthermore, we describe the experiments used to determine the learning and forgetting curve in our environment. Our findings are compared to the theoretical models and based on this we deduce which models best describe learning and forgetting in our automotive environment. The resulting models are used to develop an adaptive help system for a speech dialogue system. The system provides only relevant context specific information.

1 Introduction

Modern premium class vehicles contain a large number of driver information and driving assistance systems. Therefore the need for enhanced display and control concepts arose. BMW's iDrive is one of these concepts, allowing the driver to choose functions by a visual-haptic interface (see Fig. 1) (Haller, 2003). In Addition to the visual-haptic interface, iDrive includes a speech dialogue system (SDS) as well. The SDS allows the driver to use a large number of functions via speech commands (Hagen et al., 2004). The system offers a context specific help function that can be activated by uttering the keyword 'options'. The options provide help in the form of a list, containing speech commands available in the current context (see dialogue 1). Currently neither the driver's preferences nor his knowledge is taken into consideration. We present a strategy to optimize the options by adaption that takes preferences and knowledge into account.

Figure 1: iDrive controller and Central Information Display (CID)

Our basic concern was to reduce the driver's memory load by reducing irrelevant information. An adaptive help system based upon an individual user model could overcome this disadvantage. In (Komatani et al., 2003) and (Libuda and Kraiss, 2003), several adaptive components can be included to improve dialogue systems, e.g. user and content adaption, situation adaption and task adaption. Hassel (2006) uses adaption to apply different dialogue strategies according to the user's experience with the SDS. In our system we concentrate on user modeling and content adaption.

In this paper, we present studies concerning learning and forgetting of speech commands in automotive environments. The results are used to develop a model describing the driver's knowledge in our SDS domain. This model is used to adapt the content of the options lists.

Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, pages 1–8,
Sydney, July 2006.

Dialogue 1

User:	”Phone.”
System:	”Phone. Say dial name, dial number or name a list.”
User:	”Options.”
System:	”Options. Say dial followed by a name, for example ’dial Alex’, or say dial name, dial number, save number, phone book, speed dialing list, top eight, last eight, accepted calls, missed calls, active calls and or or off.”

2 Learning of Commands

In this section, we determine which function most adequately describes learning in our environment. In the literature, two mathematically functions can be found. These functions help to predict the time necessary to achieve a task after several trials. One model was suggested by (Newell and Rosenbloom, 1981) and describes learning with a *power law*. Heathcote et. al. (2002) instead suggest to use an *exponential law*.

$$T = B \cdot N^{-\alpha} \quad \text{(power law)} \qquad (1)$$

$$T = B \cdot e^{-\alpha \cdot N} \quad \text{(exponential law)} \qquad (2)$$

In both equations T represents the time to solve a task, B is the time needed for the first trial of a task, N stands for the number of trials and α is the learning rate parameter that is a measure for the learning speed. The parameter α has to be determined empirically. We conducted memory tests to determine, which of the the two functions best describes the learning curve for our specific environment.

2.1 Test Design for Learning Experiments

The test group consisted of seven persons. The subjects’ age ranged from 26 to 43 years. Five of the subjects had no experience with an SDS, two had very little. Novice users were needed, because we wanted to observe only novice learning behaviour. The tests lasted about one hour and were conducted in a BMW, driving a predefined route with moderate traffic.

Each subject had to learn a given set of ten tasks with differing levels of complexity (see table 1). Complexity is measured by the minimal necessary dialogue steps to solve a task. The tasks were not directly named, but explained in order not to mention the actual command and thus avoid any influence on the learning process. There was no help allowed except the options function. The subjects received the tasks one by one and had to search for the corresponding speech command in the options. After completion of a task in the testset the next task was presented. The procedure was repeated until all commands had been memorized. For each trial, we measured the time span from SDS activation until the correct speech command was spoken. The time spans were standardized by dividing them through the number of the minimal necessary steps that had to be taken to solve a task.

2.2 Results

In general, we can say that learning takes place very fast in the beginning and with an increasing amount of trials the learning curve flattens and approximates an asymptote. The asymptote at $T_{\min} = 2\text{s}$ defines the maximum expert level, that means that a certain task can not be completed faster.

The resulting learning curve is shown in Fig. 3. In order to determine whether equation (1) or (2) describes this curve more exactly, we used a chi-squared goodness-of-fit test (Rasch et al., 2004). The more χ^2 tends to zero, the less the observed values (f_o) differ from the estimated values (f_e).

$$\chi^2 = \sum_{i=1}^{k} \frac{(f_o - f_e)^2}{f_e} \qquad (3)$$

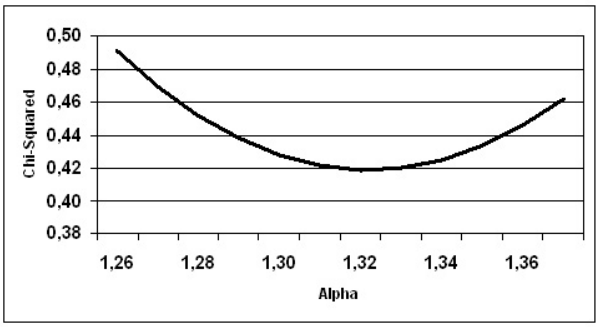
According to Fig. 2, the power law has a minimum ($\chi^2_{\min} = 0.42$) with a learning rate parameter of $\alpha = 1.31$. The exponential law has its minimum ($\chi^2_{\min} = 2.72$) with $\alpha = 0.41$. This means that the values of the exponential law differ more from the actual value than the power law’s values. Therefore, we use the power law (see Fig. 3(a)) to describe learning in our environment.

3 Forgetting of Commands

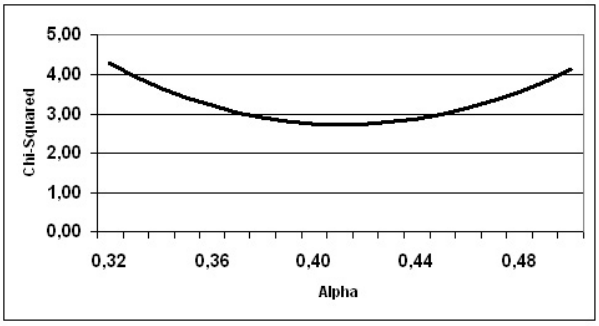
The second factor influencing our algorithm for the calculation of options is forgetting. If a command was not in use for a long period of time, we can assume that this command will be forgotten. In this section, we determine how long commands are being remembered and deduce a function most adequately describing the process of for-

Task 1	Listen to a radio station with a specific frequency
Task 2	Summary of already used destinations
Task 3	Enter a new destination
Task 4	Start navigation
Task 5	Turn off speech hints
Task 6	3D map
Task 7	Change map scale
Task 8	Avoid highways for route calculation
Task 9	Turn on CD
Task 10	Display the car's fuel consumption

Table 1: Tasks for learning curve experiments



(a) χ^2 for the Power Law



(b) χ^2 for the Exponential Law

Figure 2: Local χ^2 Minima

getting in our environment. In (Rubin and Wenzel, 1996) 105 mathematical models on forgetting were compared to several previously published retention studies. The results showed that there is no generally appliable mathematical model, but a few models fit to a large number of studies. The most adequate models based on a logarithmic function, an exponential function, a power function and a square root function.

$$\mu_{\text{new}} = \mu_{\text{old}} \cdot \ln(t+e)^{-\delta} \quad \text{(logarithmic)} \quad (4)$$

$$\mu_{\text{new}} = \mu_{\text{old}} \cdot e^{-\delta \cdot t} \quad \text{(exponential)} \quad (5)$$

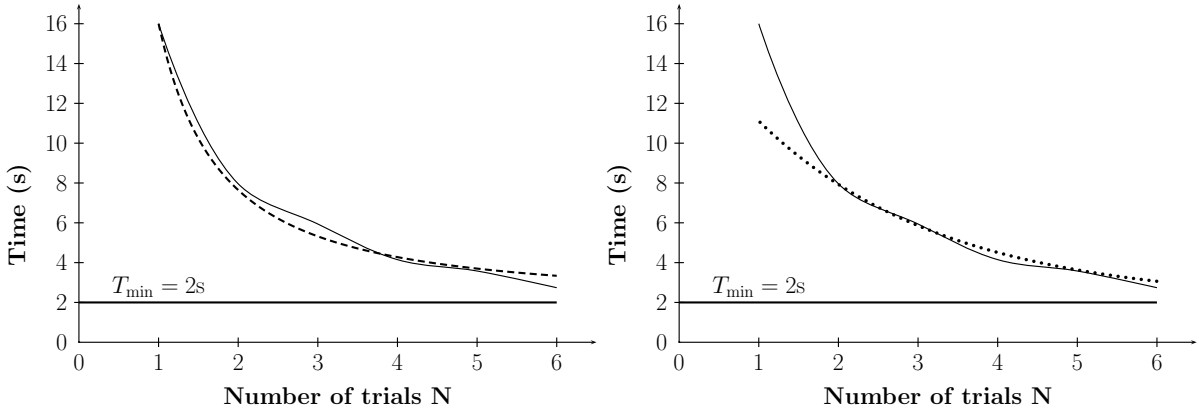
$$\mu_{\text{new}} = \mu_{\text{old}} \cdot (t+\delta)^{-\delta} \quad \text{(power)} \quad (6)$$

$$\mu_{\text{new}} = \mu_{\text{old}} \cdot e^{-\delta \cdot \sqrt{t}} \quad \text{(square root)} \quad (7)$$

The variable μ represents the initial amount of learned items. The period of time is represented through t while δ defines the decline parameter of the forgetting curve. In order to determine the best forgetting curve for SDS interactions, we conducted tests in which the participants' memory skills were monitored.

3.1 Test design for forgetting experiments

The second experiment consisted of two phases, learning and forgetting. In a first step ten subjects learned a set of two function blocks, each consisting of ten speech commands (see table (2)). The learning phase took place in a BMW. The tasks and the corresponding commands were noted on



(a) Observed learning curve and power law (dashed) with $\alpha = 1.31$

(b) Observed learning curve and exponential law (dotted) with $\alpha = 0.42$

Figure 3: Learning curves

Function block 1		Function block 2	
Task 1	Start CD player	**Task 11**	Turn on TV
Task 2	Listen to CD, track 5	**Task 12**	Watch TV station 'ARD'
Task 3	Listen to tadio	**Task 13**	Regulate blowers
Task 4	Listen to radio station 'Antenne Bayern'	**Task 14**	Change time settings
Task 5	Listen to radio on frequency 103,0	**Task 15**	Change date settings
Task 6	Change sound options	**Task 16**	Change CID brightness
Task 7	Start navigation system	**Task 17**	Connect with BMW Online
Task 8	Change map scale to 1km	**Task 18**	Use phone
Task 9	Avoid highways for route calculation	**Task 19**	Assistance window
Task 10	Avoid ferries for route calculation	**Task 20**	Turn off the CID

Table 2: Tasks for forgetting curve experiments

a handout. The participants had to read the tasks and uttered the speech commands. When all 20 tasks were completed, this step was repeated as long as all SDS commands could be freely reproduced. These 20 commands built the basis for our retention tests.

Our aim was to determine how fast forgetting took place, so we conducted several memory tests over a time span of 50 days. The tests were conducted in a laboratory environment and should imitate the situation in a car if the driver wants to perform a task (e.g. listen to the radio) via SDS. Because we wanted to avoid any influence on the participant's verbal memory, the intentions were not presented verbally or in written form but as iconic representations (see Fig. 4). Each icon represented an intention and the corresponding speech command had to be spoken.

Intention $\longrightarrow$ Task $\longrightarrow$ Command $\longrightarrow$ Success

Icon $\longrightarrow$ Task $\longrightarrow$ Command $\longrightarrow$ Success

Figure 4: Iconic representation of the functions: phone, avoid highways and radio

This method guarantees that each function was

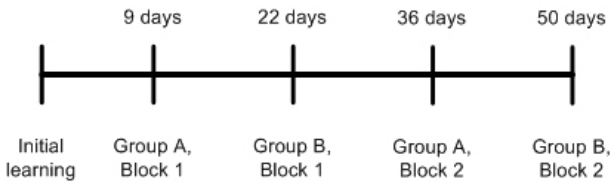
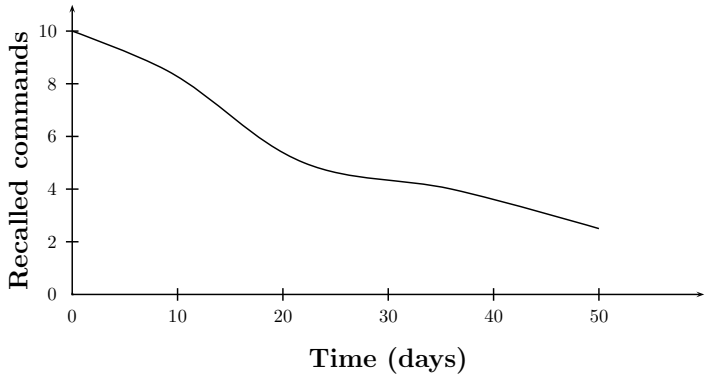
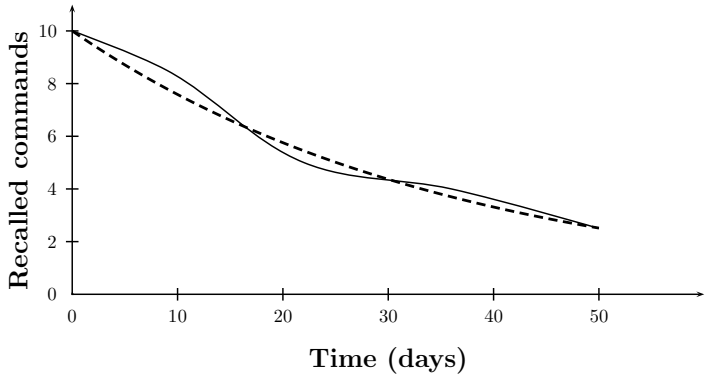


Figure 5: Test procedure for retention tests

only used once and relearning effects could not influence the results. As a measure for forgetting, we used the number of commands recalled correctly after a certain period of time.



(a) Empirical determinded forgetting curve



(b) Exponential forgetting curve (dashed) with $\delta = 0.027$

Figure 6: Forgetting curves

3.2 Results

The observed forgetting curve can be seen in Fig. 6(a). In order to determine whether equation (4), (5), (6) or (7) fits best to our findings, we used the chi-squared goodness-of-fit test (cf. section 2.2). The minima χ^2 for the functions are shown in table (3). Because the exponential function (see Fig. 6(b)) delivers the smallest χ^2, we use equation (5) for our further studies.

Function	χ^2	Corresponding δ
logarithmic	2.11	0.58
exponential	0.12	0.027
power	1.77	0.22
square root	0.98	0.15

Table 3: χ^2 values

Concerning forgetting in general we can deduce that once the speech commands have been learned, forgetting takes place faster in the beginning. With increasing time, the forgetting curve flattens and at any time tends to zero. Our findings show that after 50 days about 75% of the original number of speech commands have been forgotten. Based on the exponential function, we estimate that complete forgetting will take place after approximately 100 days.

4 Providing Adaptive Help

As discussed in previous works, several adaptive components can be included in dialogue systems, e.g. user adaption (Hassel and Hagen, 2005), content adaption, situation adaption and task adaption (Libuda and Kraiss, 2003). We concentrate on user and content adaption and build a user model.

According to Fischer (2001), the user's knowledge about complex systems can be divided into several parts (see Fig. 7): well known and regularly used concepts (*F1*), vaguely and occasionally used concepts (*F2*) and concepts the user believes to exist in the system (*F3*). *F* represents the complete functionality of the system. The basic idea behind the adaptive help system is to use information about the driver's behaviour with the SDS to provide only help on topics he is not so familiar with. Thus the help system focuses on *F2*, *F3* within *F* and finally the complete functionality *F*.

For every driver an individual profile is gen-

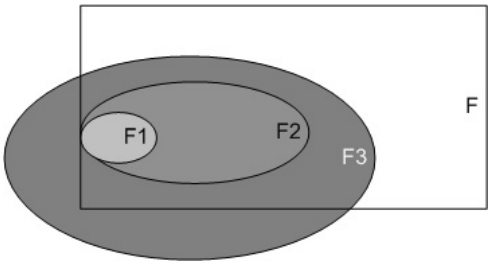


Figure 7: Model about the user's knowledge on complex systems

erated, containing information about usage frequency and counters for every function. Several methods can be used to identify the driver, e.g. a personal ID card, a fingerprint system or face recognition (Heckner, 2005). We do not further focus on driver identification in our prototype.

4.1 Defining an Expert User

In section 2 we observed that in our environment, the time to learn speech commands follows a power law, depending on the number of trials (N), the duration of the first interaction (B) and the learning rate parameter (α). If we transform equation (1), we are able to determine the number of trials that are needed to execute a function in a given time T.

$$N = \sqrt[-\alpha]{\frac{T}{B}} \tag{8}$$

If we substitute T with the minimal time $T_{\min}$ an expert needs to execute a function ($T_{\min} = 2$s, cf. section 2.2), we can estimate the number of trials which are necessary for a novice user to become an expert. The only variable is the duration B, which has to be measured for every function at its first usage.

Additionally, we use two stereotypes (novice and expert) to classify a user concerning his general experience with the SDS. According to Hassel (2006), we can deduce a user's experience by monitoring his behaviour while using the SDS. The following parameters are used to calculate an additional user model: help requests *h* (user asked for general information about the system), options requests *o* (user asked for the currently available speech commands), timeouts *t* (the ASR did not get any acoustic signal), onset time *ot* (user needed more than 3 sec to start answering) and barge-in *b* (user starts speech input during the system's speech output). The parameters are noted in a vector $\vec{UM}$.

The parameters are differently weighted by a weight vector $\vec{UM}_w$, because each parameter is a different indicator for the user's experience.

$$\vec{UM}_w = \begin{pmatrix} h & = & 0.11 \\ o & = & 0.33 \\ t & = & 0.45 \\ ot & = & 0.22 \\ b & = & -0.11 \end{pmatrix} \tag{9}$$

The final user model is calculated by the scalar product of $\vec{UM} \times \vec{UM}_w$. If the resulting value is over a predefined threshold, the user is categorized as novice and a more explicit dialogue strategy is applied, e.g. the dialogues contain more expamples. If the user model delivers a value under the threshold, the user is categorized as expert and an implicit dialogue strategy is applied.

4.2 Knowledge Modeling Algorithm

Our findings from the learning experiments can be used to create an algorithm for the presentation of the context specific SDS help. Therefore, the option commands of every context are split into several help layers (see Fig. 8). Each layer contains a

Layer 1		Layer 2		Layer 3	
Item A	1	Item E	5	Item I	9
Item B	2	Item F	6	Item J	10
Item C	3	Item G	7	Item K	11
Item D	4	Item H	8	Item L	12

Figure 8: Exemplary illustration of twelve help items divided into three help layers

maximum of four option commands in order to reduce the driver's mental load (Wirth, 2002). Each item has a counter, marking the position within the layers. The initial order is based on our experience with the usage frequency by novice users. The first layer contains simple and frequently used commands, e.g. dial number or choose radio station. Complex or infrequent commands are put into the lower layers. Every usage of a function is logged by the system and a counter i is increased by 1 (see equation 10).

Besides the direct usage of commands, we also take transfer knowledge into account. There are

several similar commands, e.g. the selection of entries in different lists like phonebook, adressbook or in the cd changer playlists. Additionally, there are several commands with the same parameters, e.g. radio on/off, traffic program on/off etc. All similar speech commands were clustered in functional families. If a user is familiar with one command in the family, we assume that the other functions can be used or learned faster. Thus, we introduced a value, σ, that increases the indices of all cammnds within the functional families. The value of σ depends on the experience level of the user.

$$i_{\text{new}} = \begin{cases} i_{\text{old}} + 1 & \text{direct usage} \\ i_{\text{old}} + \sigma & \text{similar command} \end{cases} \quad (10)$$

In order to determine the value of σ, we conducted a small test series where six novice users were told to learn ten SDS commands from different functional families. Once they were familiar with the set of commands, they had to perform ten tasks requiring similar commands. The subjects were not allowed to use any help and should derive the necessary speech command from their prior knowledge about the SDS. Results showed that approximately 90% of the tasks could be completed by deducing the necessary speech commands from the previously learned commands. Transferring these results to our algorithm, we assume that once a user is an expert on a speech command of a functional family, the other commands can be derived very well. Thus we set $\sigma_{expert} = 0.9$ for expert users and estimate that for novice users the value should be $\sigma_{novice} = 0.6$. These values have to be validated in further studies.

Every usage of a speech command increases its counter and the counters of the similar commands. These values can be compared to the value of N resulting from equation (8). N defines a threshold that marks a command as known or unknown. If a driver uses a command more often than the corresponding threshold ($i > N$), our assumption is that the user has learned it and thus does not need help on this command. It can be shifted into the lowest layer and the other commands move over to the upper layers (see Fig. 9).

If a command is not in use for a long period of time (cf. section 3.2), the counter of this command steadily declines until the item's initial counter value is reached. The decline itself is based on the results of our forgetting experiments (cf. section 3.2) and the behaviour of the counter is described by equation (5).

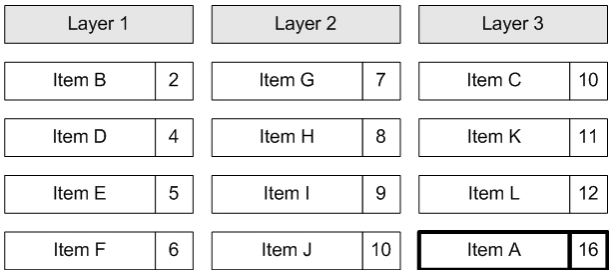


Figure 9: Item A had an initial counter of $i = 1$ and was presented in layer 1; after it has been used 15 times (i > N), it is shifted into layer 3 and the counter has a new value $i = 16$

5 Summary and Future Work

In this paper we presented studies dealing with learning and forgetting of speech commands in an in-car environment. In terms of learning, we compared the power law of learning and the exponential law of learning as models that are used to describe learning curves. We conducted tests under driving conditions and showed that learning in this case follows the power law of learning. This implies that learning is most effective in the beginning and requires more effort the more it tends towards an expert level.

Concerning forgetting we compared four possible mathematical functions: a power function, an exponential function, a logarithmic function and a square root function. Our retention tests showed that the forgetting curve was described most adequately by the exponential function. Within the observed time span of 50 days about 75% of the initial amount of speech commands have been forgotten.

The test results have been transferred into an algorithm specifying the driver's knowledge of commands within the SDS. Based on the learning experiments we are able to deduce a threshold that defines the minimal number of trials that are needed to learn a speech command. The forgetting experiments allow us to draw conclusions on how long this specific knowledge will be remebered. With this information, we developed an algorithm for an adaptive options list. It provides help on unfamiliar speech commands.

Future work focuses on usability tests of the prototype system, e.g. using the PARADISE evaluation framework to evaluate the general usabil-

ity of the system (Walker et al., 1997). One main question that arises in the context of an adaptive help system is if the adaption will be judged useful on the one hand and be accepted by the user on the other hand. Depending on user behaviour the help system could shift its contents very fast, which may cause some irritation. The test results will show whether people get irritated and whether the general approach for the options lists appears to be useful.

References

Gerhard Fischer. 2001. User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11:65–86.

Eli Hagen, Tarek Said, and Jochen Eckert. 2004. Spracheingabe im neuen BMW 6er. ATZ.

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

Liza Hassel and Eli Hagen. 2005. Evaluation of a dialogue system in an automotive environment. In *6th SIGdial Workshop on Discourse and Dialogue*, pages 155–165, September.

Liza Hassel and Eli Hagen. 2006. Adaptation of an Automotive Dialogue System to Users Expertise and Evaluation of the System.

Andrew Heathcote, Scott Brown, and D. J. K. Mewhort. 2002. The Power Law Repealed: The case for an Exponential Law of Practice. *Psychonomic Bulletin and Review*, 7:185–207.

Markus Heckner. 2005. Videobasierte Personenidentifikation im Fahrzeug – Design, Entwicklung und Evaluierung eines prototypischen Mensch Maschine Interfaces. Master's thesis, Universität Regensburg.

Kazunori Komatani, Fumihiro Adachi, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi Okuno. 2003. Flexible Spoken Dialogue System based on User Models and Dynamic Generation of VoiceXML Scripts. In *4th SIGdial Workshop on Discourse and Dialogue*.

Lars Libuda and Karl-Friedrich Kraiss. 2003. Dialogassistenz im Kraftfahrzeug. In *45. Fachausschusssitzung Anthropotechnik der DGLR: Entscheidungsunterstützung für die Fahrzeug- und Prozessführung*, pages 255–270, Oktober.

Allen Newell and Paul Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive skills and their acquisition*. Erlbaum, Hillsdale, NJ.

Björn Rasch, Malte Friese, Wilhelm Hofmann, and Ewald Naumann. 2004. *Quantitative Methoden*. Springer.

David Rubin and Amy Wenzel. 1996. One hundred years of forgetting: A quantitative description of retention. *Psychological Review*, 103(4):734–760.

Marilyn Walker, Diane Litman, Candace Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Associationfor Computational Linguistics*, pages 271–280, Morristown, New Jersey. Association for Computational Linguistics.

Thomas Wirth. 2002. Die magische Zahl 7 und die Gedächtnisspanne.