Identification of interactivity sequences in interactions with spoken dialog systems

Stefan Schmidt[1], *Klaus-Peter Engelbrecht*[1], *Matthias Schulz*[1], *Martin Meister*[2], *Julian Stubbe*[2], *Mandy Töppel*[2], *Sebastian Möller*[1]

[1]Deutsche Telekom Laboratories, Technische Universität Berlin, Germany
[2]Center for Technology and Society, Technische Universität Berlin, Germany

[stefan.schmidt01, Klaus-Peter.Engelbrecht, Matthias-Schulz]@telekom.de,
[meister, stubbe, toeppel]@ztg.tu-berlin.de,
Sebastian.Moeller@telekom.de

Abstract

The MeMo workbench is an existing system for semi-automated usability testing and following a rule based approach. The rules which are actually used have been obtained from usability expert knowledge as well as from empirical observations in a more or less uncontrolled way. In this paper we present a sociotechnical approach for finding rules, which implement probable user behavior, in a structuralized and more realistic way. We will give short introduction to the MeMo workbench followed by the description of our approach and the resulting experiment.

Index Terms: interactivity sequences, semi-automated usability testing, user task, rules, memo

1. Introduction

The MeMo Workbench (MeMo) is a system for semi-automated usability evaluation of interactive systems [1]. MeMo supports the simulation of interactions for different kinds of user interface modalities such as voice, graphics, touch, gesture and combinations (the implementation of touch and gesture is still ongoing). The result of the user behavior simulation is the basis for the detection of usability problems and the prediction of perceived usability.

MeMo simulates the behavior of users interacting with a system on the basis of two models: A system interaction model (SIM) describing the system behavior and its internal states, and a user interaction model (UIM) describing probable user behavior towards the system. The behavior of the UIM is influenced by probabilities for choosing a particular interaction step; these probabilities are modified by rules reflecting the characteristics of the system (model) as well as user characteristics. If a rule applies, it changes the probability for a certain action to be chosen by the UIM in the actual state of the simulated dialog. In the current version of the MeMo workbench, these rules have been obtained from usability expert knowledge as well as from empirical observations in a more-or-less uncontrolled way. In this paper we follow a new path for finding rules, which implement probable user behavior in a structuralized and more realistic way.

It is our assumption that users adhering to certain user groups will show a behavior that reflects the transfer of well-learned practices towards a new systems or technology. Therefore we try to discover interactivity sequences, which are not individual, but specific for groups of users while using a spoken dialog system. We suppose the best way to discover such practices is the usage of sociological empirical observations, including video devices and in-depth interviews. We conducted an experiment under such conditions with INSPIRE[1], a Smart Home Environment [2] [3].

In the next section we describe the basic principles of MeMo, focusing on the use of rules and the enhancement of the MeMo task model by introducing sub tasks. Afterwards we will introduce the sociotechnical approach and present our experiment, followed by a discussion of the first results.

2. Modeling process

As described above, MeMo uses a system- and a user interaction model. The UIM is part of the MeMo itself, whereas the system model has to be created by a person (the model developer) with expertise in the system to be evaluated. This task of the modeling process is being supported through different tools of the workbench.

In order to evaluate a system MeMo needs information about the design of the dialogs and the logical connections between them. Both kinds of information are merged in the system model of the system to be evaluated, which is represented by a finite state machine. For each state the model developer defines the available dialog(s) and outgoing transitions to other states. Each transition has a condition and a transition action that could change variable values in the system model. The set of all transitions in the system model represents the part of the application logic describing the dialog order in the evaluated application. Thus, the system model consists of three main parts: dialogs, states and their transitions, and a freely definable variable set. The system model variables describe information to be exchanged between UIM and SIM as well as parameters controlling the application logic to be simulated.

In addition to the system- and user interaction model MeMo uses system tasks and user tasks to model concrete interactions, which should be evaluated. A system task defines a task that could be executed within the evaluated application, e.g. on an answering machine: “play a message”. Such a task is described

[1] INfotainment management with SPeech Interaction via REmote microphones and telephone interfaces

by a name and several conditions, each defined by the model developer. A condition is implemented as a Boolean expression that operates on values chosen from system model variables. After each transition from one state to another the system task's conditions are checked. If all conditions have been evaluated as true, the simulated interaction will be marked as successfully fulfilled. The information how a task could be fulfilled within the evaluated system is described by a user task, which contains three pieces of information:

1. The system task that will be evaluated.
2. A start state for the simulated interaction, chosen among all system states.
3. The user knowledge, which contains all piece of information that have to be transmitted from the UIM to SIM, in order to reach the goal given by the system task.

The user knowledge is represented by a set of Attribute Value Pairs (AVP), e.g. the AVP *(Device:answeringMachine)* means that the UIM should transfer the concept *answeringMachine*, if the SIM (thus the simulated system) asks for a device.

This combination of system task and user task allows the description of simple, single tasks like "play a message from the answering machine". In this case the user task (U_1) has the knowledge (K_1) to fulfill the system task T_m, starting from the SIM state S_n. K_1 contains two pieces of information as AVPs: use the device *answeringMachine* (*am*) and perform the action *play* message. The formalization of these facts is listed below.

$$\begin{gathered} \text{let } n, m \in \mathbb{N} \\ U_1 = (K_1, S_n, T_m) \\ K_1 = ((Device, am), (Action, play)) \\ U_1 = (((Device, am), (Action, play)), S_n, T_m) \end{gathered}$$

3. Rule based evaluation

In this section we describe the rule-based approach on which the workbench is grounded. The rule engine is one of the main components of the workbench, which affects the simulated behavior of the virtual user and the modeled system by rules.

The JESS rule engine[2] is used as the rule engine for the workbench and initialized with specific workbench rules. The rules are called Decision Probability Prediction Rules (DPPR's) and they are formalized descriptions of typical user behavior. All facts used in the DPPR are derived from empirical observations [4]. To make the DPPR's easy processable and extensible all rules are stored as XML-Files. MeMo automatically transfers changes in the XML file corpus into the JESS format, which is readable by the rule engine.

All rules used for the workbench consist of a declaration, a condition and a consequence. The declaration contains general information – such as name or description of the rule – and an arbitrary count of element groups. Element groups are buttons or other user interface elements on which the rule can be applied. The condition section is optional and describes the values of the user model's properties or values for the properties of an interaction element, which have to be fulfilled to apply the rule (see Table 1). If a rule does not have a condition the consequence is always applied.

Type	Description
	Example
Dynamic user attributes	Can be changed during the interaction.
	irritation, frustration
Static user attributes	Stable during the interaction.
	eye-sight, age, affinity to technology
Dialog attributes	Describe dialog properties.
	button size, prompt length
Interaction properties	Dynamic interaction properties.
	duration, sequent no matches

Table 1: The four attribute/properties types used in MeMo rules.

The consequence section describes how the probabilities for declared element groups or the user's intention, and the values of the dynamic user attributes have to be changed. A value can be changed absolutely or relatively. Absolute changes will overwrite the previous value, whereas relative changes modify the current value by a certain amount.

MeMo uses four types of rules, whereby each rule type changes the probability of occurrence for various parameters in the interaction. The first rule type is called Interaction Rules. Interaction Rules refer to potential interaction elements and contribute to the determination of the interaction probability distribution matrix. A textual example to make it more clear what an interaction rule can stand for: If the user has limitations in visual perception and the button has a poor contrast and a small size, then the probability that the user will perceive and press the button is reduced.

The second kind of rules, the History Rules, compares the current system state with the intended system state and with the recent interaction history in order to modify the dynamic user attributes. Thus, if the user model is searching for specific information and can't find the information in a certain amount of time or a certain number of interaction-steps a specific rule will increase the level of frustration.

Intention Rules are the third type of rules used in the workbench. Intention Rules affect the current goal of the user model. Most of the time and especially at the beginning of a simulation, the user model tries to fulfill the main task goal. If this goal is not achieved within a certain time period or after a certain number of interaction steps, there is a chance that the user model changes temporary his intention to ask for help instead of fulfilling the global goal.

The last kind of rules is called AVP Rules. These rules affecting the number of concepts communicated in a single interaction step. This is especially important for the evaluation of speech dialog systems.

Because some rules affect the processing of other rules, the rule engine is called four times to change different decision matrices by applying rules. The first call will change dynamic user attributes by applying rules for static user attributes and History Rules. The second run will change the user intention by using Intention Rules and also rules for static user attributes. The purpose of that run is to decide, if the user model will quit, ask for help or continue as before. The third iteration will apply AVP Rules to calculate the number of attribute value pairs/concepts specified by the UIM in the current simulation step. Finally, the last run of the rule engine calculates the potential next interaction step. In this rule engine step static user attributes, in-

[2] http://www.jessrules.com

teraction rules and the user intention are used to calculate the probability distribution matrix for all possible interactions at the current system state.

4. A socio-technical approach

Sociological research on technology usage can roughly be divided into three research fields: First, acceptance research that postulates specific attributes of a technology and asks, which individual or situative factors influence if a technology is used/accepted or not. Second, demographic approaches focusing on social factors like gender, education, age or income and thus try to explain usage among different social groups. And thirdly, approaches of the more recent sociology of technology that aim at reconstructing typical patterns – or practices – of man-machine-interactivity among users. The approach taken in this project can be ranked among the third group. The user model that is integrated into the MeMo workbench is based on typified "ways of conduct" or practices which are reconstructed from the empirical observation of interactivity sequences in several usability evaluations.

Successful technology usage is usually performed in the form of incorporated practices or routines. The user transfers those practices or routines from previous experiences with a familiar technology into situations that confront the user with a unknown type of technology. It is assumed that users accumulate a certain implicit "know-how" of interacting with technology. That "know-how" is gained through the everyday use of different applications and is "brought into" a new situation. Hence, it becomes a challenge from the user perspective to apply this "know-how" and transfer it into a successful performance with the confronting system. On the other hand usability problems are a matter of adapting specific everyday ways of conduct. These ways of interacting with a technology, which on the one hand are embodied by users through a high degree of continuity, but on the other hand allow situative flexibility and adaptivity, is what we understand as practices of technology usage.

In order to understand the way users adopt a specific technology, these practices have to be reconstructed. Usually this happens through methodologically controlled observations of man-machine-interactivity in its full empirical scope. This ethnographic research tradition is followed and extended by the research program of Technographics [5] by adding the formative medial and structuring power of artifacts [6]. This allows not only to analyze the details of individual man-machine-interactivity, but also to identify general usage problems and generate conclusions for the design process [7].

However, the social practice-approach of analyzing user actions does not neglect that demographic factors matter in people's behavior towards technology. Not in the sense that age, education or income influence technology usage per se, but rather that different age groups embody different practices depending on mainstream technologies of a given time. Sackmann and Weymann investigated this relation between chronological thrusts of innovation and the timeliness of technology possession, experience and competence of different cohorts [8]. Within their concept of "Technology Generations", they state that after the successfully introducing a technology on the market especially younger cohorts gained specific competences of handling the new technology quicker than older (ibid.: 64). An example from the launching of Automated Teller Machines (ATM) shows that generations for whom the computer "is part of everyday life" adapted the use of ATMs earlier than older generations, who remained to prefer personal bank service instead. Sackmann and Weymann conclude that younger generations are less reserved when it comes to stepping into a "dialog" with these machines, because they are more familiar with the operative logic of ATMs, which is similar to computer products of the digitalization (ibid.: 65).

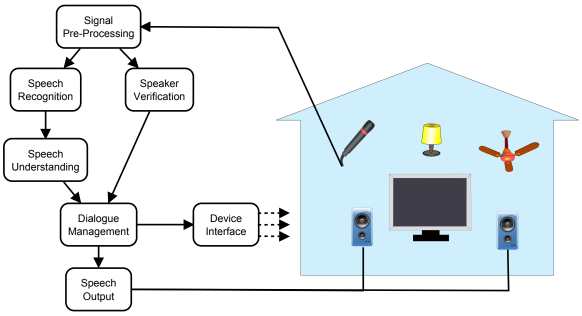

Figure 1: Schematic presentation of the INPSIRE system [3]

5. Experimental set-up

We conducted an usability experiment with 33 participants (20 – 76 year, mean age 44 (STD 16.21), 15(45.5%) male, 18(54.5%) female) with different educational level(academic and professional education, as well as students). Each test participant was asked to carry out three tasks with different technical applications provided by the spoken dialog system INSPIRE [2] [3]. This smart home environment enables the user to operate different home appliances (here MP3 player, electronic program guide (EPG), answering machine) by an interactive speech dialog. The MP3 Player and the EPG support the user with graphical outputs on the television screen, e.g. the program survey in the EPG. The original system consists of a pre-processing module, a speech recognizer, a speech understanding module, a speaker verification module, a dialog manager, a device interface (controls the home appliances), and a speech output module[3]. Figure 1 shows a schema of the overall structure. As we were not interested in the impact of speech recognition errors, a Wizard-of-Oz substituted the speech recognizer and the speaker verification in the test set-up.

Each participant was filmed during the interaction and the television screen was recorded as well. All inputs from the wizard, outputs from the text-to-speech engine and internal system events were logged. Furthermore a sociologist observed the test user on a monitor and notes striking actions and reactions on an observation sheet, e.g. posture, reusing commands or canceling the task. After the participants has interacted with the system they answered a usability questionnaire [3], which was developed in accordance with ITU-T Rec. P.581 [9]. Directly after filling in the questionnaires, a sociologist did an in-depth interview with the participant (duration ~25 minutes), which was based on the observation sheet and the recorded interaction. The interviewer confronted the participant with a short video screening of sequences from his/her own behavior and asked him/her to comment on it. A Digit Span Test [10] (forward and backward) completed the interview and tested the participant's concentration, attention and auditory short memory. The results might be used to construct appropriate user groups.

In the interaction between a participant and INSPIRE, we

[3] MARY Text-to-Speech System, see: `http://mary.dfki.de`

try to discover the practices well-learned by acting with other technologies. Due to this fact we used circumscribing instructions on the exercise sheet. As an example follows the exercise for the answering machine task:

> Check all message on your answering machine, please. Try to call-back your grandmother and delete the message from the caller who has misdialed. If your grandmother's telephone connection is busy, please continue the task. When you're finished, please say ”INSPIRE exit“.

Such a task description minimizes the take over of correct voice commands from the instruction sheet and does not force a special order of user actions. Thus, the user was forced to use practices learned before. We used the command ”INSPIRE exit“ as the signal for the test participant's decision to end the task.

6. Discussion and first results

6.1. The interactivity model

A specific challenge of the socio-methodological approach was the creation of an undisturbed environment that encouraged people to freely apply their intuitive strategies of interacting with the system and capturing these moments on video. The resulting sequences have served as our main source of analysis, which were complemented by in-depth interviews including video feedbacks. The stimulus created by the video feedback has shown great effect on creating constructive reflections by the test participant and brought up specific individual intentions that were relevant for interacting with the system. By confronting a person with his/her own behavior it was possible to reconstruct some impacts on certain ways of acting and what practices he or she had tried to apply in a specific situation.

The interactivity patterns once being identified, they can be formalized in terms of rules and implemented into MeMo. This requires the translation of qualitative empirical results into a format, which is suitable for the MeMo Workbench and imply a reduction of complexity of our empirical data.

In order to generate an output allowing the distinction of typical patterns that are significant for specific user groups we created a “bottom-up” model from our data that consists of elementary interactivity steps. From the sociological perspective this means the creation of a human counterpart of the system model, which contains elements that correspond to system states as well as specific prompts from the speech dialog system.

One of our main concerns for the model was the integration of performative “ways of conduct” – practices – as well as basic cognitive processes that coordinate the action undertaken by the user. Our interactivity model (see Figure 2) is based on “elementary actions” that pile up to interactivity sequences. This means that a sequence may contains different elementary actions in contrast to static models, where user attributes are modeled as given and won't change within one run of simulation. The elementary actions of our model contain three basic elements: task, operationalization and appraisal. These elements are empirical categories, which can be used to code and analyze the data:

Task A task constitutes what a user would like to do with the system. In our model we distinguish two different types of tasks: manipulation and information. “Manipulation” are commands that shall bring the system into a new state. “Information” are commands that are directed to the retrieval of information about the system or the available options in a specific state.

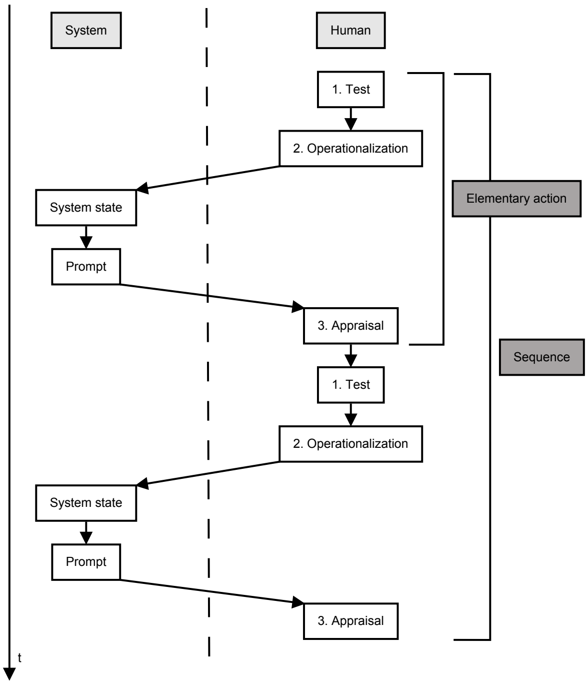

Figure 2: Sequence diagram of the interactivity model

Operationalization This element corresponds to the idea of practices and the integration of a “way of doing things” in our model. Hence, the empirical data will be coded according to how people performed the tasks they have posed in their command. Therefore we distinguish buzzword, subset and sentence. A “buzzword” is a very precise command that consists of one word only. “Subsets” are commands that are articulated in a very accentuated way (“machine language”) like buzzwords, but contain at least two parts and are thus less precise. “Sentences” are user commands that orientate on social communication, including forms of politeness and filler words.

Appraisal This cognitive process is a reaction to the system prompt, in which the previous task and operationalization are evaluated to the effect if the system has fulfilled the command appropriately or not. We differentiate between “expected” and “unexpected” system behavior. In our analysis we considered the full scope of the video data including gestures and mimics of the users.

The empirical codes form cascading elementary actions that pile up to interactivity sequences. The resulting patterns can be interpreted for each case and be brought into relation with specific user attributes. This could for instance be the deployment of steps that have been learned from handling a different kind of ICT and now proof to be adaptable to the INSPIRE system.

The analysis of the experiment's data is still in progress, but we have taken a first hypotheses from the patterns of interactivity sequences. Our hypotheses is that users who first employ “tasks” directed to accessing information about the system and then in a second step pose tasks of manipulation, based on the accessed information, have less interaction problems than those

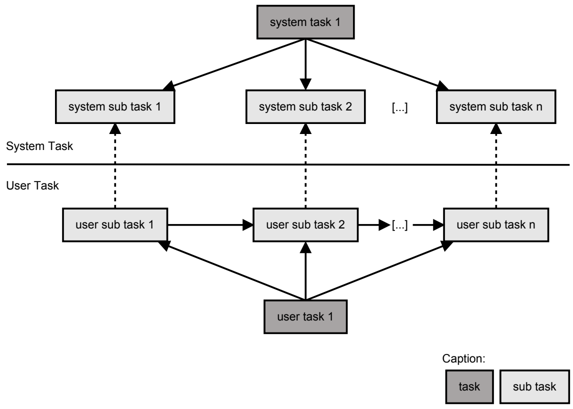

Figure 3: Relations between tasks and sub tasks

users, who pose tasks of system manipulation directly from the start of the interaction. The first kind of user behavior requires a sophisticated understanding of the operative logic of digital menus. Such an experience can only be gained through the everyday use of comparable devices – and the embodiment of practices of operating pertinent technologies.

6.2. Modeling user sub tasks

As written in the introduction and exemplified in Section 4 we try to discover interactivity sequences. Hence, we created complex tasks for the experiment. Our first results while evaluating the interactivity model shows, that the participants uses sequences of actions, which could not be modeled with the existing user task model (see Section 2). An example for a complex task with the answering machine is shown in Figure 5

Such interactions with a speech dialog system can be modeled and simulated with MeMo by introducing the concept of sub tasks. By adapting the idea of ConcurTaskTrees [11] each task (system or user) might have several sub tasks of the same type and with the same properties, including the feature to own sub tasks. Thus, it is possible to create trees of system or user tasks. Whereas a tree of system tasks only defines a set of unordered goal conditions, the order of user sub tasks in a tree specifies their order of execution (see Figure 3). The figure shows additionally that each user task needs a system task describing the goal condition for the task. If the condition is becoming true, the UIM have has successfully interacted. An example how to model the task "play the first message, then play the second message and delete the message" with user sub tasks is shown in Figure 4.

In [12] and [13] Schatzmann presented an agenda-based user simulation. The concept of "knowledge" that is used in MeMo's user task also works with attribute value pairs, but unlike Schatzmann's agenda, the piece of information have no transfer order. The MeMo user model tries to guess what information is suitable for the actual dialog and after that it tries to send this information to the SIM. Which information could be suitable depends on the prompt shown by the system model. However, as described above it is possible to define an order for the information to be transferred by the use of user sub tasks.

6.3. Simulation with user sub task

The user instructions used in the experiment do not impose a special order of actions, but on the other hand we need an ordered sequence of actions for the definition of an initial user task, including user sub tasks, for the simulation in MeMo. Therefore we trained a Markov chain with all sequences of actions observed for the answering machine in the experiment. Thus we got the most probable order of actions for the answering machine task (see Figure 5). Based on this chronology it was possible to define a user task with MeMo, which owns one user sub task for each identified action.

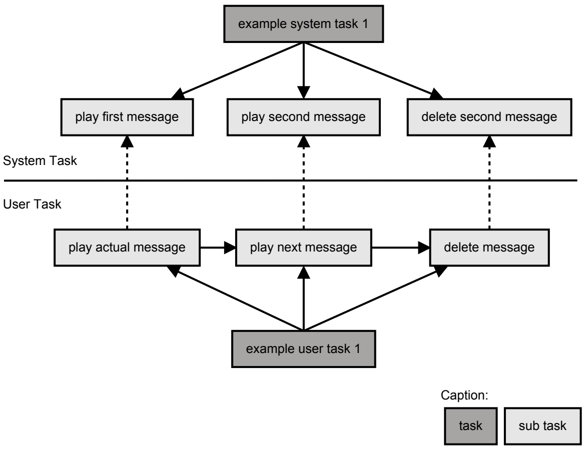

Figure 4: Answering machine example for the usage of sub tasks.

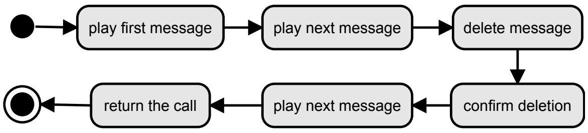

Figure 5: Most probable chronology of user actions with the answering machine in the experiment.

In order measure the influence of the new rules to the simulation success, it is necessary to capture the baseline of simulation results produced with the actual state of MeMo. Thus, we have built models for each of the three INSPIRE applications used in our experiment. Tables 2 and 3 list the first results in terms of interaction parameters for the answering machine application.

Experiment (N = 31) [absolute (relative)]	Simulation (N = 200) [absolute (relative)]
23 (74.2%)	187 (93.5%)

Table 2: Tasks success rate (answering machine) from experiment and simulation.

The experimental dataset contains 31 records (out of 33) that are valid, the MeMo simulation runs for 200 iterations, whereby each iteration simulates one complete course of dialog. Table 2 shows that the task success rate of the simulated users deviates strongly from the rate in the experiment. As shown in Table 3 The simulation predicts the number of user turns, the concept error rate (CER) and the dialog duration in a good approximation to the experiment. The CER represents the mean ratio of wrongly transmitted AVPs in relation to all used AVPs,

Parameter	Experiment (N = 31) [mean (STD)]	Simulation (N = 200) [mean (STD)]
Turns	10.45 (3.11)	10.37 (1.87)
AVP	1.53 (.27)	1.46 (.11)
Dialog Duration (ms)	208.68 (74.82)	200.69 (43.13)
Deletions	1.06 (1.46)	1.38 (1.17)
Insertions	.29 (.59)	.09 (.32)
Substitutions	.35 (1.02)	.06 (.26)
CER	.1 (.12)	.09 (.07)
No Match	1.16 (1.27)	1.84 (.98)

Table 3: Results of collected interaction parameters. The data come from INSPIRE's answering machine application and the MeMo simulation of that application. (CER: Concept Error Rate, AVP: Attribute Value Pair)

each per user and completed dialog. A concept error could be a deletion, an insertion or a substitution. The main factor for the simulation's CER are deletions. Insertions and substitutions in the simulation have less influence then in the real user test. Whether the Wizard-of-Oz, the application model or another factor are the cause for the differences is object of our current analysis. Furthermore we are still analyzing the reason for the smaller standard deviation (STD) of each parameter in the simulation and also the influence of this effect on the evaluation results.

7. Conclusion and future work

The presented interactivity model can be used to formalize interactivity sequences contained in qualitative data which were ascertained in our experiment.

In the frame of a sociotechnical approach it was necessary to define complex tasks for the experiment, in order to discover interactivity pattern. The implementation of user sub tasks for the modeling process made the simulation of such complex tasks with the MeMo workbench possible.

Simulations with user sub tasks provide promising results, but their still existing deviations between predicted interaction values and those from a real use experiment. Mainly the recall of simulated interactions, among all interactions used by real users, have to be increased. Therefore we will formalize new rules with our described approach.

8. Acknowledgments

The work has been carried out in the frame of an ongoing interdisciplinary research project (UserModel, project number 10033294) funded by the Deutsche Forschungsgemeinschaft (DFG) at the Quality and Usability Lab and the Center for Technology and Society, both Technische Universität Berlin.

9. References

[1] K.-P. Engelbrecht, M. Kruppa, S. Möller, and M. Quade, "MeMo workbench for semi-automated usability testing," in *Proceedings of Interspeech 2008: 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22–26*, 2008, pp. 1662–1665.

[2] S. Möller, J. Krebber, A. Raake, P. Smeele, M. Rajman, M. Melichar, V. Pallotta, G. Tsakou, B. Kladis, A. Vovos, J. Hoonhout, D. Schuchardt, N. Fakotakis, T. Ganchev, and I. Potamitis, "Inspire: Evaluation of a smart-home system for infotainment management and device control," *CoRR*, vol. cs.HC/0410063, 2004.

[3] S. Möller, P. Smeele, H. Boland, and J. Krebber, "Evaluating Spoken Dialogue Systems According to De-Facto Standards: A Case Study," *Computer Speech and Language*, vol. 21, pp. 26–53, November 2007.

[4] S. Möller, R. Englert, K. Engelbrecht, V. Hafner, A. Jameson, A. Oulasvirta, A. Raake, and N. Reithinger, "MeMo: Towards automatic usability evaluation of spoken dialogue services by user error simulations," in *Proceedings of Interspeech 2006—ICSLP: 9th International Conference on Spoken Language Processing, Pittsburgh, PA, USA*, 2006, pp. 1786–1789.

[5] W. Rammert and C. Schubert, Eds., *Technografie. Zur Mikrosoziologie der Technik*. Frankfurt, New York: Campus Verlag, 2006.

[6] W. Rammert, "Technikvergessenheit der Soziologie? Eine Erinnerung als Einleitung." in *Technik und Sozialtheorie*, F. am Main: Campus, Ed. Fankfurt am Main: Werner Rammert, 1998, pp. 9–28.

[7] L. Suchmann, J. Blomberg, J. E. Orr, and R. Trigg, "Reconstructing technologies as social practice," *American Behavioral Scientist*, vol. 43, pp. 392–408, 1999.

[8] R. Sackmann and A. Weymann, *Die Technisierung des Alltags. Generationen und technische Innovationen*. Campus, 1994.

[9] S. Möller, "Subjective Quality Evaluation of Telephone Services Based on Spoken Dialogue Systems," International Telecommunication Union, Geneva, Switzerland, ITU-T Recommendation P.851, November 2003, based on ITU-T Contr. COM 12-59 (2003).

[10] M. von Auster, A. Neubauer, and R. Horn, Eds., *WIE Wechsler Intelligenztest für Erwachsene*. Harcourt Test Services, 2006.

[11] F. Paternò, C. Mancini, and S. Meniconi, "Concurtasktrees: A diagrammatic notation for specifying task models." in *INTERACT*, ser. IFIP Conference Proceedings, S. Howard, J. Hammond, and G. Lindgaard, Eds., vol. 96. Chapman & Hall, 1997, pp. 362–369. [Online]. Available: http://dblp.uni-trier.de/db/conf/interact/interact1997.html#PaternoMM97

[12] J. Schatzmann, B. Thomson, and S. Young, "Statistical user simulation with a hidden agenda," in *8th SIGDial Workshop on Discourse and Dialogue*, Antwerp, Belgium, 2007, pp. 273–282.

[13] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, "Agenda-based user simulation for bootstrapping a POMDP dialogue system," in *NAACL '07: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2007, pp. 149–152.