Analysis of Communication Failures for Spoken Dialogue Systems

Sebastian Möller [1], *Klaus-Peter Engelbrecht* [1], *Antti Oulasvirta* [1,2]

[1] Deutsche Telekom Laboratories, Berlin University of Technology, Germany
[2] Helsinki Institute for Information Technology, Helsinki University of Technology, Finland

sebastian.moeller@telekom.de, klaus-peter.engelbrecht@telekom.de, antti.oulasvirta@hiit.fi

Abstract

Communication failures are typical for interactions with spoken dialogue systems, in particular when dialogues get less structured and less foreseeable. In this paper, we adopt a new classification scheme of communication failures and their consequences and show its usefulness in three respects: (1) For the systematic analysis of data collected in user testing, (2) for the prediction of user-perceived quality and usability, and (3) for the automatic testing of usability in a simulation testbed. Experimental results are presented for two spoken dialogue systems which differ in their dialogue structure and complexity. They show that the failure classification may uncover the causes of interaction problems between user and system, irrespective of system complexity, and that failure consequences can serve as a predictor of user satisfaction.

Index Terms: spoken dialogue system, evaluation, communication failure, usability, smart home

1. Introduction

Communication failures occur frequently in the interaction with spoken dialogue systems. They stem from an asymmetry of the capabilities of the user and the system (e.g. with respect to speech recognition and understanding), from misconceptions of those capabilities, or from a wrong "mental model" of the user, i.e. a wrong conception of the system and its internal structure. Such misconceptions are common because speech-based systems are accessible to the human only indirectly, from their prompts and reactions to user speech.

The resulting communication failures are frequently called "user errors", although usually *no fault can be given to the user of the system*. We use the term "user error" more broadly here for all types of deviations from a somehow "ideal" path through a task-directed interaction. By definition, user errors lead to a non-optimal progress (stagnation, regression, or only partial progress) towards the goal of the interaction. It is obvious that analyzing errors according to this definition is possible only if the goal of the user is known, something which might not be possible in the context of non-task-oriented systems like entertainment applications.

The analysis of such user errors is important in order to localize system deficiencies and optimize the interaction design, as it has been shown e.g. in [1]. Several classification schemes have been proposed to support such an analysis. For example, Bernsen and Dybkjær [1] distinguish 8 types of errors observed during interactions with the Danish flight information system, like ignoring system feedback, responding to questions different from the ones asked by the system, change requests through comments, the user asking questions at unforeseen states of the interaction, thinking-aloud, etc. Each error could be classified with respect to its symptoms, a diagnosis was made, and a preventive measure was proposed. On the system side, Constantinides and Rudnicky [3] classified errors along the lines of a fishbone diagram, where each bone corresponds to a particular part of the system. Bohus and Rudnicky [2] analyzed errors and recovery strategies in a system for reserving conference rooms. They focused on non-understanding errors – cases in which the system is unable to assign an interpretation to an utterance of the user – and identified low-level speech recognition and audio segmentation problems to account for 65% of the problems. They also classified out-of-application (users referring to entities or functions outside the system domain) and out-of-grammar errors (non-conformity with the systems grammar or lexicon).

We proposed an own classification scheme in [11] which was influenced by the cited classifications and by error analysis in human factors and cognitive ergonomics. This scheme was further developed to cover a wider range of systems, and optimized to be equally applicable to the analysis of interaction problems, to the automatic evaluation of systems, and to the prediction of usability. Because of this, we prefer a classification scheme which (a) addresses the user's behavior in response to the one of the system, and (b) is not based on the underlying reasons of the problems (in terms of system components, such as [3]), but on the surface form of the utterances exchanged between user and system. The original classification scheme has been extended to generalize across systems [4], and the updated scheme will be briefly presented in Section 2.

We applied this scheme to experimental data collected with two systems which differ in their dialogue structure and complexity: The INSPIRE system as a spoken dialogue interface to domestic devices, and the BoRIS system as a telephone-based interface to a database. Details on the systems and the experiments are given in Section 3. First, we use the scheme for analyzing system deficiencies, on different levels (goal, task, command, concept and recognition), see Section 4. Then, we calculate correlations between frequencies of interaction problems and consequences on the one hand, and subjective judgments on quality and usability on the other, see Section 5. We also predict subjective judgments on the basis of error occurrences. We then propose to use the classification scheme in an automatic usability prediction approach, see Section 6. Finally, we discuss the main advantages of our approach in Section 7.

2. Error classification

For each task, we assume that there are one or several optimum solution paths, "hidden" from the user's immediate perception. The execution of the task involves commanding the system and interpreting the cues in its response so as to

transform the initial state of the system to the hidden goal state. An *error* is defined as any deviation from the optimal solution path(s). We currently distinguish five *categories of errors*:

- *Goal-level error* (capability): The system does not posses the function or capability assumed in the user's request. E.g., asking the system to control an object that is not in the system, at a level of granularity not possible by the system, etc.
- *Task-level error*: The user does not understand how to reach the goal in the interaction with the system. E.g., the command is not valid in this state of the dialog, the command is valid but does not progress or regress the dialog, the command exceeds the control capabilities of the system, or an attribute is referred to using a wrong value class (e.g. string instead of number).
- *Command-level error* (vocabulary and grammar): The user makes use of linguistic variations (synonyms, grammar, etc.) which are not understood by the system. E.g., no input is provided, some vocabulary of the utterance is not known to the system, some vocabulary of the utterance is misunderstood by the system, the vocabulary is not understood in the grammatical construction utilized by the user, or a grounding error occurs (no mutual belief or common conception is reached).
- *Concept-level error* (modeling): Issuing a command that would be valid if the system represented the "world" in a different way. It is possible to imagine another kind of model/categorization of the world in which this utterance would not constitute an error.
- *Recognition-level error*: Complementary to the above classes of "user errors", this class captures interaction failures which even a completely cooperative user cannot influence, e.g. ASR errors or wizard mistyping.
- *Other*: All errors not classifiable in the above classes.

The errors may result in the following *consequences*:

- *Stagnation*: The system takes the user to a prompt that is as close to the task goal as the previous prompt, i.e., the goal can still be reached with as many steps as before. A special case of this are called *Repetition*: The system repeats the same prompt.
- *Regression*: The system goes to a state that is farther away from the task goal than the previous state. A special case of this is called *Restart*: The system returns to its initial state, loosing any progress achieved.
- *Partial progress*: The system goes to a state which is closer to the task goal, but not all the information in the user's utterance is processed.
- *Complete progress*: No negative consequence occurs.

3. Experimental data

We applied the error classification scheme to interactions collected in two Wizard-of-Oz (WoZ) experiments with two different spoken dialogue systems: Experiment 1 with a speech-based interface to domestic appliances, with a less-structured dialogue, and experiment 2 with a typical telephone-based system for information access. A transcribing wizard replaced the speech recognizer in both experiments to address dialogue-related problems in more detail, and to avoid that errors mainly stem from the ASR component alone. Details on the experiments are given in the literature [8][10], and only a brief summary will be given here. In addition, we will further describe the error annotation procedure.

3.1. Exp. 1: INSPIRE smart-home system

This experiment has been carried out in the frame of the EU-funded IST project INSPIRE. The respective system provides speech control over a number of domestic appliances (TV, video recorder, electronic program guide, 3 lights, blinds, fan, answering machine) through a unified dialogue structure. The dialogue manager allows for mixed initiative, and the system parses also incorrectly-formulated sentences and proceeds on the basis of incomplete information.

24 participants (10 f, 14 m, average age 23.7 years) were asked to solve 3 scenarios in a simulated home environment, each including 9-11 tasks on different devices. Following each interaction with the system, the participants rated 37 different statements referring to different quality aspects. The questionnaire was designed according to ITU-T Rec. P.851 [5], and details on the experiment can be found in [10].

3.2. Exp. 2: BoRIS restaurant information system

The BoRIS system allows searching for restaurants in the town of Bochum, Germany, through a telephone-based spoken dialogue. It is implemented as a finite-state machine and optionally uses an explicit confirmation strategy. A transcribing wizard replaced the speech recognizer during the experiment, but artificial errors were generated on the wizard's transcription, in order to simulate recognition rates between 60 and 100% (see [8] for details). Speech output has been generated via pre-recorded messages, speech synthesis, or a combination of both.

40 participants (11 f, 29 m, mean 29 years) interacted five times with the BoRIS system over a simulated phone connection in an office environment, following five different scenarios. Each scenario involved one task for obtaining restaurant information. Following each interaction with the system, the participants rated 26 statements referring to different quality aspects on a questionnaire according to ITU-T Rec. P.851 [5]. Details on the experiment can be found in [8].

3.3. Annotation of errors and consequences

The interactions of both experiments have been logged, transcribed and annotated by a human labeler. The categories have first been defined on the basis of the INSPIRE system, and later adjusted to also reflect the characteristics of the BoRIS system. For each exchange of information between user and system, the labeler could set one or more error labels, and one consequence label. Because the user might have produced more than one error in each utterance, the error categories are not mutually exclusive, but supportive. 2343 exchanges have been labeled in exp. 1, and 2415 in exp. 2.

Labeling reliability was tested on an independent set of 290 exchanges obtained with the INSPIRE system, but not being part of exp. 1. An outside coder was hired and trained to use the same coding scheme in several training sessions. In coding Cohen's κ (a statistical measure of inter-coder agreement) between the first and the second coder, we found that task-level, command-level and recognition-level reliability was high ($\kappa \in [0.70;0.91]$, 0.60 being considered as an appropriate threshold for claiming substantial inter-rater agreement [7]), whereas the reliability of goal-level and representation-level errors was lower ($\kappa \in [0.34;0.44]$). We think that the low frequency of errors in these classes in the test data set is responsible for this finding. In turn, the consequences could all be annotated with a high reliability ($\kappa \in [0.60;0.97]$). Overall, we think that the coding reliability is sufficiently high to justify further analysis of both error classes and consequences.

4. Error analysis

The coding reveals that 28% of the exchanges in exp. 1 and 32% in exp. 2 contained at least one error. Some of the errors in exp. 2 are due to the deliberate introduction of recognition errors in exp. 2, see Table 1. Without the recognition errors, still 19% of erroneous utterances remain. Thus, the coding scheme leads to a significant number of errors in both cases.

Table 1. *Relative frequencies (%) of errors in the data set. Percentages do not sum up to 100% because of multiple errors per utterance.*

Exp.	Goal	Task	Repres.	Comm.	Recog.	Other
1	1	44	11	51	1	1
2	24	15	6	22	43	1

In exp. 1, most errors occur on the command and the task level. Command-level errors are mostly related to out-of-vocabulary words or phrases (37% of all errors), 6% stem from grammar problems. Task-level errors consist of the user stating a concept at a state where this is not possible, coding the concept incorrectly (e.g. name instead of number), or specifying two logically-related items in one user utterance, e.g. "switch on *two* lamps". This type of error may have been triggered by some macros included in the system, allowing the user to "switch on *all* lamps". This may have led the user to over-estimate the capabilities of the system. Most of the representation-level errors are in the domain of space, the utterance representing the location of the lamps in a way different from that of the system.

In exp. 2, most of the user-related errors occur on the goal and the command level. Goal-level errors mainly come from external restrictions (no fitting restaurants are contained in the database, as was foreseen in the scenario), and command-level errors from out-of-vocabulary words.

The relative frequencies of the consequences of errors are displayed in Table 2. "Complete progress" was annotated only for exp. 2 when no valid restaurant could be found in the database, which was a case foreseen by the scenario.

Table 2. *Relative frequencies (%) of consequences.*

Exp.	Stagnation	Regression	Partial progress	Complete progress
1	50	12	38	-
2	30	28	15	27

The effect of errors should be reflected in a lower efficiency, and potentially also a lower task success rate. In fact, a correlation analysis of the exp. 2 data shows that the efficiency-related parameters dialogue duration (*DD*) and the number of turns (*#Turns*) correlate significantly ($p<0.05$) and strongly with stagnation, repetition and regression consequences (Pearson correlation $r\in[0.59;0.74]$), and negatively with the subjective impression of the dialogue being short ($r\in[-0.32;-0.40]$). With respect to effectiveness, the frequency of regression and restart correlates significantly and negatively with the subjective impression of whether the system provided the desired information, and with an expert annotation of task success ($r\in[-0.20;-0.47]$). For exp. 1, correlations between *DD* or *#Turns* and all non-progressive consequences were significant and positive ($r\in[0.39;0.78]$, except between *#Turns* and partial progress). All *DD*, *#Turns* and the subjective judgments related to effectiveness were correlated with negative error consequences. Thus, the detrimental effect of errors is supported by this analysis.

A further analysis shows that many errors can only be avoided at the expense of other errors. For example, widening the number of slots (concepts) which can be filled at each state will open initiative to the user and may avoid task-level errors, but may reduce speech recognition and understanding accuracy. The errors also helped to detect conceptual differences between the user and the system model; they showed that users wished to specify device locations in a different way than it is foreseen by the INSPIRE system.

5. Impact on subjective judgments

The detrimental effect of errors should also be reflected in the subjective judgments. Because a large number of judgments were collected in both experiments, there is a high chance that correlations appear due to chance alone. We therefore only report on highly significant ($p<0.01$) correlations between error and consequence frequencies on the one hand, and (a) the user's rating on overall quality, OVQ, or (b) the arithmetic mean over all subjective judgments after aligning the positive/negative statements, MEANQ, on the other.

For exp. 2, highly significant correlations to both OVQ and MEANQ were found for the recognition-level errors ($r\in[-0.28;-0,29]$) and the goal-level errors ($r\in[-0.24;-0.26]$). The task-level errors still had a significant correlation with OVQ ($r=-0.22$). For exp. 1, the other error classes seem to be more decisive for subjective quality: Command-level errors ($r\in[-0.42;-0.53]$) and representation-level errors ($r\in[-0.33;-0.41]$) had highly significant correlations with both OVQ and MEANQ. This shows that the impact of a particular type of error may be specific to the system under investigation.

In turn, the correlations between OVQ/MEANQ and the error consequences are more stable across experiments. Stagnation had a highly significant correlation in both experiments ($r\in[-0.25;-0.41]$), as well as the particular case of a repetition ($r\in[-0.22;-0.53]$). In addition, the regression consequence had a highly significant effect in exp. 2 ($r\in[-0.33;-0.34]$) and is still significant ($p<0.05$) in exp. 1 ($r\in[-0.25;-0.26]$). For exp. 2, also the specific case of system restart ($r\in[-0.23;-0.25]$) is highly significant.

The values show that both error frequencies and consequences are correlated with the user's perception of overall system quality. The relationships are not very strong, but they are consistent and meaningful. Whereas individual correlations test only the impact of one particular variable on quality, multivariate regression models capture the linear effect of a multitude of variables. The PARADISE framework proposed by Walker et al. [12] can be used to predict the impact of a number of parameters quantifying the interaction and system performance on "user satisfaction". Such parameters include the number of words per system turn (*WPST*), system turn duration (*STD*), system response delay (*SRD*), and measures of task success (*TS* and κ). Details on the parameters can be found in [6].

For exp. 2, we calculated multivariate regression models for the target variable MEANQ. Input to the models are either 28 interaction parameters collected from a log-file annotation after the experiment (IP), or the error and consequence frequencies (EC), or both. All input parameters are normalized to a zero mean and unity standard variation, square roots are calculated from the EC parameters to reduce the effects of Poisson-like distributions, and a stepwise inclusion of input parameters was selected in order to reduce the model size. Table 3 shows the resulting models.

Table 3. *Regression models for exp. 2.*

Input	Parameters	R^2_{adj}	E_p
EC	- 0.32·#Regr. + 0.20·#Partial Progr. - 0.19·#Stagn - 0.16·#Restart	0.243	0.861
IP	1.34·*WPST* - 0.39·*#Turns* -1.64·*STD* + 0.78·*SRD* + 0.25·*TS* - 0.21·κ	0.471	0.881
EC+IP	- 0.38·#Regr. + 1.30·*WPST* - 1.69·*STD* + 0.86·*SRD* - 0.19·#Restart	0.477	0.865

The results show that the corrected variance covered by the models (R^2_{adj}) is larger for the interaction parameters compared to the error and consequence frequencies alone. Still, a small improvement (in terms of higher R^2_{adj}) can be reached when including regression and restart frequencies in the models, however at the expense of a slightly higher prediction error E_p. Apparently, the frequency of negative error consequences can help to increase the accuracy of prediction models like PARADISE. Still, there is no single predictor of usability, neither for the EC nor for the IP input parameters.

6. Simulation of user behavior

The error classification scheme is currently been used for generating "erroneous" user behavior in an automatic evaluation approach which is described in [9]. Our approach is to set up a model for the behavior of an ideal user, in the sense of a user that follows the "optimum path" through the interaction. From this "optimum path", deviations are produced by generating errors according to our classification scheme.

The generation of goal-level errors can be handled on the level of the task description: The user (model) asks for a task which is not supported by the current system version. The definition of such tasks requires (a) knowledge of the tasks which are supported by the system, and (b) some domain knowledge about similar tasks which would be logic in the given domain. The generation of task-level errors is easy when the system model is described in terms of a state machine, as it is common practice in commercial dialogue systems. Here, a task-level error may simply be generated by issuing a user command which does not fit to the state the system is in. The generation of command-level errors is currently implemented by means of a synonym dictionary, from which synonyms are grabbed for the concepts understood by the system. User utterances are generated on the basis of the system grammar, taking the in-vocabulary words as well as the synonyms as composites. Work is ongoing to check how realistic user utterances generated this way are.

7. Discussion and Conclusions

In this paper, we propose a classification of interaction failures for the analysis and optimization of spoken dialogue systems. We use the scheme to classify errors observed with two spoken dialogue systems of different complexity, a simple telephone-based system and complex smart-home system.

The results show that about 99% of all interaction failures observed by an external annotator could be classified according to 5 classes. The classes generalize across systems, and support an in-depth analysis of the user's behavior towards the system. The patterns of frequent types of errors were notably different for the two systems. They show the limitations of the systems under test and may help to select trade-offs between system characteristics, because not all errors can be avoided without provoking other types of errors.

The frequency of errors, and in particular the consequences of the errors, correlate with user judgments on quality and usability. The correlations are weak (up to -0.53) but statistically significant and meaningful. This makes us confident that the classification scheme can be used for quality and usability prediction, by using consequence frequencies as an input to linear regression models such as PARADISE.

We also showed that the classification scheme is useful for generating "erroneous" user behavior in an automatic model-based evaluation approach. Further work is necessary to show how realistic behavior generated according to the scheme is, compared to collected human-machine dialogues.

8. Acknowledgements

Exp. 1 was supported by the EC-funded IST project INSPIRE (IST-2001-32746) and by the MeMo project funded by Deutsche Telekom AG. The authors would like to thank all colleagues who supported the experiments and annotation, as well as Anthony Jameson for his comments and suggestions.

9. References

[1] Bernsen, N. O., Dybkjær, H. and Dybkjær, L., Designing *Interactive Speech Systems. From First Ideas to User Testing.* Springer Verlag, 1998.

[2] Bohus, D., Rudnicky, A., "Sorry, I didn't catch that! – An investigation of non-understanding errors and recovery strategies", in: *Proc. 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, 2-3 Sept., 128-143, 2005.

[3] Constantinides, P.C., Rudnicky, A.I., "Dialog Analysis in the Carnegie Mellon Communicator", in: *Proc. 6th Europ. Conf. on Speech Communication and Technology (Eurospeech'99)*, Budapest, 1:243-246, 1999.

[4] Engelbrecht, K.-P., *Fehlerklassifikation und Benutzbarkeits-Vorhersage für Sprachdialogsysteme auf der Basis von mentalen Modellen*, Magister thesis, Deutsche Telekom Labs, TU Berlin, 2006.

[5] ITU-T Rec. P.851, *Subjective Quality Evaluation of Telephone Services Based on Spoken Dialogue Systems*, International Telecommunication Union, Geneva, 2003.

[6] ITU-T Suppl. 24 to P-Series Rec., *Parameters Describing the Interaction with Spoken Dialogue Systems*, International Telecommunication Union, Geneva, 2005.

[7] Landis, J., Koch, G., "The Measurement of Observer Agreement for Categorical Data", *Biometrics* 33:159-174, 1977.

[8] Möller, S., *Quality of Telephone-Based Spoken Dialogue Systems*, Springer, New York NY, 2005.

[9] Möller, S., Englert, R., Engelbrecht, K., Hafner, V., Jameson, A., Oulasvirta, A., Raake, A., Reithinger, N., "MeMo: Towards Automatic Usability Evaluation of Spoken Dialogue Services by User Error Simulations", in: *Proc. 9th Int. Conf. on Spoken Language Processing (Interspeech 2006 – ICSLP)*, Pittsburgh PA, 1786-1789.

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

[11] Oulasvirta, A., Möller, S., Engelbrecht, K., Jameson, A. "The Relationship of User Errors to Perceived Usability of a Spoken Dialogue System", in: *Proc. 2nd ISCA/DEGA Tutorial and Research Workshop on Perceptual Quality of Systems*, Berlin, 61-67, 2006.

[12] Walker, M.A., Litman, D.J., Kamm, C.A. and Abella, A., "PARADISE: A Framework for Evaluating Spoken Dialogue Agents", in: *Proc. ACL/EACL 35th Meeting*, Madrid, 271-280, 1997.