WHAT SHOULD YOUR SPEECH SYSTEM SAY TO ITS USERS, AND HOW?

Guidelines for the Design of Spoken Language Dialogue Systems

Niels Ole Bernsen, Hans Dybkjær and Laila Dybkjær
Centre for Cognitive Science, Roskilde University
PO Box 260, DK-4000 Roskilde, Denmark
emails: nob@cog.ruc.dk, dybkjaer@cog.ruc.dk, laila@cog.ruc.dk
Phone: +45 46 75 77 11 Fax: +45 46 75 45 02

Abstract

As telephone-based spoken language dialogue systems are becoming commercially available, the need for improved design, development and evaluation methods and tools has become apparent. The design of dialogue models for such systems is still based mainly on common sense, experience and intuition, and trial and error, rather than on established design guidelines. The paper presents a comprehensive set of principles for the design of spoken human-machine dialogue and describes their consolidation through several forms of validation. Potentially, at least, the principles can be used as dialogue design guidelines. The issues that remain in turning the principles into an established, quasi-complete body of dialogue design guidelines of certified practical usefulness are discussed.

Keywords: Spoken language dialogue design, cooperativity, design guidelines.

1. Introduction

Since the early 1990s, telephone-based spoken language dialogue systems (SLDSs) products have been entering the market place at a rapidly increasing pace [1]. Traditionally, SLDSs have been developed through the Wizard of Oz (WOZ) simulation technique in which a human (the wizard) simulates the system to be developed in dialogue with users who are made to believe that they interact with a real system [2]. The dialogues are recorded, transcribed and analysed, and the results are used to improve the dialogue model. Several WOZ iterations enable major improvements to be made on dialogue structure and contents of the system's utterances prior to implementation and subsequent testing. WOZ is costly but its use has so far been justified through the comparatively higher cost of having to revise an already implemented SLDS whose dialogue turned out to be seriously flawed. As the cost of rapid prototyping decreases, implement-test-and-revise methods are likely to gain ground in SLDS design, primarily for simple, small-vocabulary dialogues.

However, with or without WOZ and for both simple and more complex dialogue tasks, there is a need for guidelines to support dialogue design and help remove dialogue problems as early as possible in the design life-cycle. Today's dialogue model design for SLDSs is based primarily on

common sense, the individual designer's experience and intuition, and trial and error, rather than on established dialogue design principles. If the dialogue designers are not both very careful and lucky, many problems of interaction may still remain to be discovered during implementation and test of the system. A sound and comprehensive set of dialogue design guidelines might serve as an effective and systematic development and evaluation tool during early design. This could significantly reduce development time by reducing the need for lengthy WOZ experimentation, controlled user testing, and field trial cycles, thereby reducing overall development costs. The call for such guidelines is not new. Baber [3] reviewed Grice's maxims of conversation [4] and Schneidermann's "8 golden rules" of human-computer interaction design [5]. He concluded that it is not obvious how to use the former as design guidelines and that general rules such as the latter lack the clarity and specificity needed to support SLDS design. In this paper, we present a set of SLDS design principles which have been validated in three ways: through WOZ dialogue design problem-solving during the design of an advanced SLDS, i.e. the Danish dialogue system, through comparison with the Gricean maxims, and through a user test of the implemented system. Based on these validations, we hope that the proposed principles will be useful to SLDS designers and that they may provide a sound basis for further progress in expressing a complete, consistent and practically useful set of SLDS design guidelines. It seems likely that the principles cover most, if not all, aspects of dialogue design and hence could be used in the design and evaluation of the many SLDSs which are making their way from research laboratories through field testing to product development.

The Danish dialogue system may be briefly described as follows. This prototype addresses the domain of domestic airline ticket reservation. It has been developed in collaboration with the Center for PersonKommunikation, Aalborg University and the Centre for Language Technology, Copenhagen. The system runs on a PC with a DSP board and is accessed over the telephone. It is a walk-up-and-use application. The system understands speaker-independent continuous spoken Danish with a vocabulary of about 500 words and uses system-directed dialogue. The prototype runs in close-to-real-time. It has the following main modules: a speech recogniser, a parser, a dialogue module, a database, and an output module which generates concatenated pre-recorded speech [6]. The system is representative of advanced state-of-the-art systems. Comparable SLDSs are found in, e.g. [7, 8].

In what follows, Section 2 provides a link between SLDS design guidelines and principles of cooperative system dialogue. Section 3 presents our principles and describes how they were developed. Section 4 briefly compares the principles with Grice's maxims of cooperativity in human-human dialogue. Section 5 describes how the principles were validated through the user test of the implemented system. Section 6 concludes and discusses the relationship between principles of cooperative system dialogue and practically useful guidelines for SLDS design.

2. Dialogue Cooperativity

It is argued in this section that there is a direct link between principles supporting cooperative system dialogue behaviour and guidelines for SLDS dialogue design.

Current SLDS design is subject to many constraints on the dialogue between user and system. These constraints are partly derived from the technology, partly from dialogue engineering limitations and partly from an insufficient theoretical basis of dialogue design. Yet it is possible to design fully usable or habitable SLDSs for certain classes of tasks. The key to successful dialogue design, we claim, is to ensure *adequate dialogue cooperativity on the part of the system*. It is a well-established fact that task-oriented SLDS technologies are based on the assumption of cooperative *user* dialogue behaviour [9]. This fact does not, however, pose much of a problem for dialogue designers because the penalty for non-cooperativity is that users fail to get their task done. There is no point in designing the dialogue for non-cooperative users who do not care if they succeed with their task or not. Indeed, this design goal is impossible to achieve in the foreseeable future. However, habitable user-system dialogue requires that the *system's* dialogue behaviour should also be cooperative. If this is not the case, penalties can be severe, ranging from users having to repeatedly initiate clarification and repair meta-communication with the system through to failing to get the task done or abandoning SLDSs technologies altogether. *Meta-communication* is communication about the dialogue itself rather than about the task domain of the dialogue, and is typically initiated for purposes of clarification and repair. We do that in human-human spoken dialogue when we say, e.g., "Please repeat - I didn't hear what you just said" (repair), "Sorry, I said 'recognise speech', not 'wreck a nice beach'" (repair), "Is 'speech recognition' the same as 'speech understanding'?" (clarification), or "What do you mean by 'red departure'?" (clarification).

In particular the speech recognition capabilities of SLDSs are still fragile [10]. Meta-communication functionality is therefore always needed to overcome the effects of system misrecognitions. Similarly, SLDSs must enable users to have the system's latest utterance repeated when they have failed to catch the point made by the system. Beyond these two unavoidable types of user-initiated repair meta-communication, however, the system should not cause the need for other kinds of clarification and repair meta-communication**.** Speaking more generally, the system should not behave in ways which decrease the likelihood that the cooperative user gets the task done. At any stage during dialogue, the cooperative user should know what to do and how to do it, without having been misled or left without guidance by a non-cooperative system. A crucial dialogue design goal, therefore, is to optimise system dialogue cooperativity in order to prevent unnecessary user-initiated clarification and repair meta-communication as well as other kinds of unexpected user dialogue behaviour with which the system cannot cope. Such unnecessary user dialogue behaviour tends to increase the demands on the system's language comprehension and dialogue management capabilities to a level beyond what is currently technically feasible. This again decreases the user's satisfaction in communicating with the system. The practical problem therefore becomes: how to design cooperative system dialogue behaviour? To our knowledge, the question of how to design cooperative system dialogue has not been addressed in any systematic way. The answer to that question appears to generate a potentially useful set of guidelines for spoken dialogue design.

3. Establishing the Principles of Cooperative System Dialogue

The dialogue model for the Danish SLDS prototype was designed by means of the WOZ technique. Seven WOZ iterations involving a total of 24 users were performed to produce the dialogue

model. The dialogue model was in the first two iterations represented as a loosely ordered set of predefined phrases but was then turned into a graph structure (a state transition network) in order to facilitate the wizard's job and add structure to the dialogue. The graph had predefined system phrases in the nodes and expected user input contents along the edges. Throughout the WOZ experiments, interaction with the system was based on scenarios, i.e. domain relevant task descriptions. The last three WOZ iterations made use of 28 scenarios which had been designed to systematically explore the system's task domain. From the seven iterations, 125 dialogues were transcribed amounting to about seven hours of spoken language dialogue.

After each iteration the transcribed dialogues were analysed and evaluated. Evaluation results were used to improve the dialogue model before the next WOZ iteration. As the dialogue model improved, we began to match the scenarios to be used in the following iteration against the current dialogue graph structure in order to discover and remove potential user problems. *Potential user problems* are problems discovered analytically by the designers when putting themselves in the place of the actual users. By contrast, *actual user problems* are problems which actually occur during user-system dialogue. Significantly, many problems were discovered analytically through the scenario-based walkthroughs of the dialogue model. This seems to be typical of dialogue model development and illustrates the potential of dialogue design guidelines which would be used for exactly this analytical purpose. In the last two WOZ iterations, we also matched the latest version of the system's dialogue model against the transcribed WOZ corpus in order to systematically detect problems of system cooperativity from the actual user problems that occurred. Each transcribed dialogue was plotted onto the dialogue model graph. Deviations from the graph indicated unexpected user or system behaviour. The deviations were marked and their causes analysed. Figure 1 shows an annotated sub-graph from WOZ6. The annotation shows that the subject expected confirmation from the system. When it became clear that the system was not going to provide confirmation (E8), the subject asked for it (S8). The following dialogue fragment provides the background for the subject's deviation from the WOZ6 dialogue model. The subject has made a change to a flight reservation. After the user has stated the desired change, the dialogue continues (E is experimenter, S is subject):

E7: Do you want to make other changes to this reservation?
S7: No, I don't.
E8: Do you want anything else?
S8: Ah no ...I mean is it okay then?
E9: [Produces an improvised confirmation of the change made.]
S9: Yes, that's fine.
E10: Do you want anything else?

From this point the dialogue finishes as expected. Analysis convinced us that the dialogue model had to be revised in order to prevent the user-initiated clarification meta-communication observed in S8, which the implemented system would be incapable of understanding. In fact, the WOZ6 dialogue model can be seen to have violated the following dialogue design principle: *Be fully explicit in communicating to users the commitments they have made.* As a result, system confirmation of changes of reservation was added to the WOZ7 sub-graph describing change of reservation.

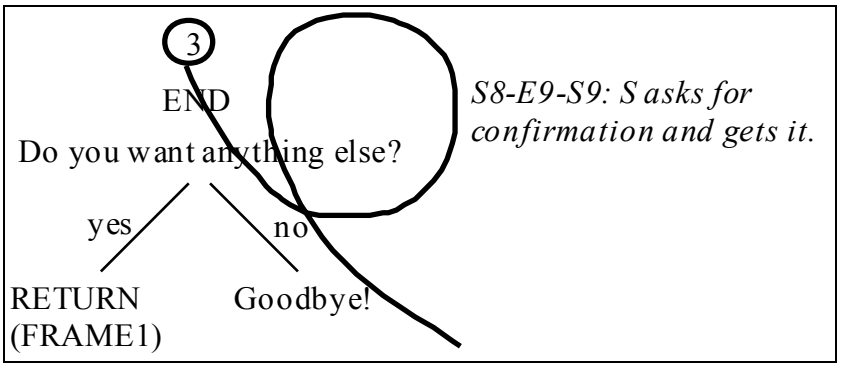

Figure 1. A plotted END of dialogue sub-graph from WOZ6. The encircled number (3) refers to the CHANGE of reservation sub-graph, cf. E7-S7 in the transcribed dialogue in the text. The user is expected to say, in S8, either "yes" or "no" to the simulated system's question (E8) "Do you want anything else?" If "yes", the system will initiate a fresh dialogue by returning to the sub-graph FRAME1. If "no", the system will end the dialogue by saying "goodbye". What actually happens is that in S8 the user initiates the clarification sub-dialogue conducted through S8-E9-S9. The boldfaced loop marks this deviation from the graph path which may reveal a dialogue design problem. The loop is annotated with numbered reference (in italics) to S8-E9-S9 and a description of the deviation. E refers to experimenter and S to subject**.**

At the end of the WOZ design phase, we did a more theoretical exercise of categorising identified dialogue design problems and expressing the corresponding dialogue design principles. We plotted the transcribed user-system dialogues from WOZ3 onwards onto their corresponding graphs. In addition, we compared each adjacent dialogue model graph pair (WOZn/WOZn+1) in order to identify and analyse all changes made to the dialogue model from WOZ3 through to WOZ7. The actual and potential problems of interaction identified in the WOZ experiments were analysed, classified and represented as violations, made by the system, of principles of cooperative dialogue. Each problem was considered a case in which the system in addressing the user had violated a principle of cooperative dialogue. The principles were made explicit based on the problems analysis. We also analysed how the system's utterances had been, or should be, improved to minimise user-initiated clarification and repair meta-communication and avoid other forms of unwanted user communication. To illustrate the WOZ corpus analysis, we present an example of an identified problem type (a) and the cooperative principle (termed 'design commitment') which has been violated (b). A justification of the principle is provided (c), followed by examples of how it was found to be violated (d). Under (d) we note whether a particular example was discovered empirically (i.e. from actual problems of interaction) or analytically (i.e. through design analysis revealing a potential problem). Finally, a solution to each problem is proposed and sometimes discussed (e). The template (a-e) was applied to each problem that had been identified.

(a) *Problem:* Non-separation between novice users who need introductory information about what the system can and cannot do and intermediate and expert users who do not need such information and for whom listening to it would only delay task performance.
(b) *Violation* of design commitment: Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).
(c) *Justification:* There are major differences between the needs of novice and expert users, one such difference being that expert users already possess the information needed to understand system functionality.

(d) *Examples:* Introduction (WOZ7): A new question was added: "Do you know this system?" First-time users may obtain additional information about the functionality of the system and about how to communicate with it. Other users may proceed directly with their task. The problem was discovered from user problems. Users complained that the system talked too much. Consideration of this complaint led to the described design improvement.
(e) *Solution:* In WOZ7 it was made optional for users to listen to the introduction to the system.

Principles	Justification
P1. Provide clear and comprehensible communication of what the system can and cannot do.	Risk of communication failure in case of lacking knowledge about what the system can and cannot do. Violation of this principle leads users to have exaggerated expectations about the system's abilities, which may lead to frustration during use of the system.
P2. Provide sufficient task domain coverage.	Risk of communication failure in case of lacking task domain information. Full task domain coverage within specified limits is necessary in order to satisfy all relevant user needs in context. Otherwise, users will become frustrated when using the system.
P3. Provide same formulation of the same question (or address) to users everywhere in the system's dialogue turns.	Need for unambiguous system response (consistency in system task performance). The principle is meant to reduce the possibility of communication error caused by users' understanding a new formulation of a question as constituting a different question from one encountered earlier.
P4. Take users' relevant background knowledge into account.	Need for adjustment of system responses to users' relevant background knowledge and inferences based thereupon. This is to prevent that the user does not understand the system's utterances or makes unpredicted remarks such as, e.g., questions of clarification, which the system cannot understand or answer.
P5. Avoid 'semantical noise' in addressing users.	Need for unambiguous system response. The design commitment is to reduce the possibilities of evoking wrong associations in users, which in their turn may cause the users to adopt wrong courses of action or ask questions which the system cannot understand.
P6. It should be possible for users to fully exploit the system's task domain knowledge when they need it.	Risk of communication failure in case of inaccessible (or not easily accessible) task domain information. In such cases, users may pose questions which the system is unable to understand.
P7. Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.	Need for adjustment to users' background knowledge and inferences based thereupon. Users may otherwise fail to understand the system.
P8. Provide clear and sufficient instructions to users on how to interact with the system.	Risk of communication failure in case of unclear or insufficient instructions to users on how to interact with the system. Users may become confused about the functionality of the system.
P9. Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).	There are major differences between the needs of novice and expert users, one such difference being that expert users already possess the information needed to understand system functionality.
P10. Avoid superfluous or redundant interactions with users (relative to their contextual needs).	Users tend to get irritated and inattentive from unnecessary system turns.
P11. Be fully explicit in communicating to users the commitments they have made.	Users need feedback from the system on the commitments made in order to assure correctness.
P12. Reduce system talk as much as possible during individual dialogue turns.	Users get bored and inattentive from too much uninterrupted system talk.
P13. Provide feedback on each piece of information provided by the user.	Immediate feedback on user commitments serves to remove users' uncertainty as to what the system has understood and done in response to their utterances.
P14. Provide ability to initiate repair if system understanding has failed.	When system understanding fails, the system should initiate repair meta-communication and not leave the initiative with the user.

Table 1. The cooperative SLDS dialogue design principles (left-hand column) and their justifications (right-hand column).

The described procedure led to the identification of 14 principles of cooperative human-machine dialogue based on analysis of 120 examples of user-system interaction problems (Table 1). If the principles were observed in the design of the system's dialogue behaviour, we assumed, this would serve to reduce the occurrence of user dialogue behaviour that the system had not been designed to handle. The table includes a justification of each principle, which serves the additional purpose of clarifying its meaning and scope. Although not explicitly stated in each justification, we take it to be straightforward that violations of any of the principles may lead users to initiate meta-communication or other non-desirable dialogue behaviour, because this is the strategy naturally adopted in human-human conversation in such cases.

4. Comparison with Grice's Theory

We had developed our principles of cooperative system dialogue independently of Grice's cooperativity theory. Having become aware of the close relevance of Grice's work, and prior to the user test, we compared the principles with Grice's Cooperative Principle (CP) and maxims [4]. In this process, the principles achieved their present formulation as shown in Tables 2 and 3 (Grice's maxims being incorporated without changes). Grice's Cooperative Principle is a general principle which says that, to act cooperatively in conversation, one should make one's "conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which one is engaged" [4]. Grice proposed that the CP can be further explicated in terms of four groups of simple maxims which are neither claimed to be jointly exhaustive nor to be mutually exclusive or non-overlapping. The maxims are marked with an asterisk in Table 2.

A detailed discussion of, and comparison with, Grice's work is presented elsewhere [11]. The main difference between Grice's work and ours is that the maxims were developed to account for cooperativity in human-human dialogue, whereas our principles were developed to account for cooperativity in human-machine dialogue. Grice's primary interest lies in the inferences which an interlocutor is able to make when the speaker *deliberately* violates one of the maxims to produce an oblique message. Our primary interest is in *non-deliberate* violations of maxims and principles. It is exactly when a human or an SLDS non-deliberately violates a maxim that dialogue problems are likely to occur. However, whether violated deliberately or non-deliberately, the principles or maxims are the same and their function remains that of achieving the shared dialogue goal as directly and smoothly as possible.

The comparison with Grice's maxims yielded a clear-cut result. It turned out that the principles include the maxims as a subset. The principles manifest aspects and principles of cooperative task-oriented dialogue which were not addressed by Grice. The distinction between *principle* and *aspect* (Table 2) is important because an aspect represents the property of dialogue addressed by a particular maxim or principle. Furthermore, the comparison made us aware of the distinction be-

tween *generic* and *specific* principles. As shown in Table 2, Grice's maxims are all generic. However, a generic principle may subsume one or more specific principles which specialise the generic principle to certain classes of phenomena. Although subsumed by generic principles, we believe that the specific principles in Table 2 are important in SLDS dialogue design (Section 6).

The demonstration that a sub-set of the principles P1-P14 in Table 1 are roughly equivalent to the maxims GP1-GP9 in Table 2, goes as follows. *P5* is a generalised version of GP6 (non-obscurity) and GP7 (non-ambiguity) and may, without any consequence other than improved clarity, be replaced by GP6 and GP7. *P6* can be considered an application of GP1 (informativeness) and GP9 (orderliness). If the system adheres to GP1 and GP9, there is a maximum likelihood that users obtain the task domain information they need from the system when they need it. The system should say enough and address the dialogue topics in an order which is as close as possible to the order expected by users. If the user expects some topic to come up early in the dialogue, that topic's non-occurrence at its expected "place" may cause a clarification sub-dialogue which the system cannot understand. In WOZ3, for instance, the system did not ask

Dialogue Aspect	GP No.	SP No.	Generic or Specific Principle	P No.
Aspect 1: **Informativeness**	GP1		*Make your contribution as informative as is required (for the current purposes of the exchange).	P6
		SP1	Be fully explicit in communicating to users the commitments they have made.	P11
		SP2	Provide feedback on each piece of information provided by the user.	P13
	GP2		*Do not make your contribution more informative than is required.	P10
Aspect 2: **Truth and evidence**	GP3		*Do not say what you believe to be false.	implicit
	GP4		*Do not say that for which you lack adequate evidence.	implicit
Aspect 3: **Relevance**	GP5		*Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.	P10
Aspect 4: **Manner**	GP6		*Avoid obscurity of expression.	P5
	GP7		*Avoid ambiguity.	P5
		SP3	Provide same formulation of the same question (or address) to users everywhere in the system's dialogue turns.	P3
	GP8		*Be brief (avoid unnecessary prolixity).	P12
	GP9		*Be orderly.	P6
Aspect 5: **Partner asymmetry**	GP10		Inform the dialogue partners of important non-normal characteristics which they should take into account in order to behave cooperatively in dialogue. Make sure that they are able to do so.	generated
		SP4	Provide clear and comprehensible communication of what the system can and cannot do.	P1
		SP5	Provide clear and sufficient instructions to users on how to interact with the system.	P8
Aspect 6: **Background knowledge**	GP11		Take partners' relevant background knowledge into account.	P4
		SP6	Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.	P7
		SP7	Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).	P9
	GP12		Take into account legitimate partner expectations as to your own background knowledge.	generated
		SP8	Provide sufficient task domain knowledge and inference.	P2
Aspect 7: **Repair and clarification**	GP13		Initiate repair or clarification meta-communication in case of communication failure.	generated
		SP9	Initiate repair meta-communication if system understanding has failed.	P14

		SP10	Initiate clarification meta-communication in case of inconsistent user input.	user test
		SP11	Initiate clarification meta-communication in case of ambiguous user input.	user test

Table 2. Principles of cooperative system dialogue. GP means generic principle. SP means specific principle. The generic principles are expressed at the same level of generality as are the Gricean maxims (marked with an *). Each specific principle is subsumed by a generic principle. The left-hand column characterises the aspect of dialogue addressed by each principle. The right-hand column shows the relationship to the principles (P) in Table 1. GP3 and GP4 were tacitly assumed. GP11, GP12 and GP13 were generated from specific principles. SP10 and SP11 were derived from the user test corpus.

users about their interest in discount. Having expected the topic to come up for some time, users therefore began to inquire about discount when approaching the end of the reservation dialogue.
P10 is virtually equivalent to GP2 (do not overdo informativeness) and GP5 (relevance). P10 may, without any consequence other than improved clarity, be replaced by GP2 and GP5. *P12* is near-equivalent to GP8 (brevity). This takes care of all of Grice's maxims except *GP3 and GP4*. Although Grice's maxims of truth and evidence (GP3, GP4) have no counterparts among our principles, these maxims may simply be included among the principles. The reason is that one does not design an SLDS in the domain of air ticket reservation which provides false or unfounded information to customers. The system must act as a perfect domain expert vis-à-vis its users. In other words, the maxims of truth and evidence are so important to the design of SLDSs that they are unlikely to emerge during dialogue design problem-solving. This is probably why we did not come across GP3 and GP4 when establishing our principles. During system implementation, one constantly worries about truth and evidence. It cannot be allowed, for instance, that the system confirms information which has not been checked with the database and which might be false or impossible.

Principles P1-P4, P7-P9, P11 and P13-P14 do not have any equivalents among the maxims. Some of these principles highlight dialogue aspects that are absent from Grice's theory which only considers the aspects of *informativeness, truth and evidence, relevance* and *manner* (Table 2). The rest are specific principles. Thus, *P1 and P8* are both specific principles (SP4 and SP5 in Table 3) subsumed by the new generic principle GP10 which highlights the importance in dialogue design of taking *dialogue partner asymmetry* into account. Dialogue partner asymmetry occurs, roughly, when one or more dialogue partners is not in a normal condition or situation. For instance, a dialogue partner may have a hearing deficiency or be located in a particularly noisy environment. In such cases, dialogue cooperativity requires the taking into account of that participant's special characteristics. For obvious reasons, dialogue partner asymmetry is important in SLDS dialogue design. The machine is not a normal dialogue partner and users have to be aware of this to avoid communication failure. Being limited in its language and task capabilities, and intended for walk-up-and-use application, our SLDS needs to protect itself from unmanageable dialogue contributions by providing users with a mental model of what it can and cannot do. It therefore transfers part of the responsibility for cooperation in dialogue onto its users. Importantly, users *must be able* to take on this responsibility. They should not be asked to conduct their dialogue with the system in ways that they cannot manage in practice.

P4 (GP11 in Table 2) introduces the dialogue aspect of *background knowledge* and of possible differences in background knowledge among different user populations and individual users. The

taking into account of users' background knowledge is important in SLDS dialogue design. In the user test, for instance, a user wanted to order a one-way ticket at discount price. The system, however, knew that discount is only possible on return tickets. It therefore did not offer the discount option nor did it correct the user's misunderstanding. At the end of the dialogue, the frustrated user asked whether or not discount had been granted, which question, of course, the system failed to understand. As long as SLDSs are unable to adaptively build substantial user models during dialogue, the user modelling task falls squarely upon the dialogue designers. *P2* (SP8, subsumed by the new GP12 in Table 2) mirrors GP11. GP12 requires the dialogue designers to make sure that the system behave as a perfect expert vis-à-vis its users within its declared domain of expertise. Ensuring that is no trivial matter. SLDS designers are continuously confronted with questions about what the system should know and what is just within, or barely outside, the system's intended or expected domain of expertise. In WOZ7, for instance, a subject expressed surprise at not having been offered the option of being put on a waiting list in a case in which a flight was fully booked. This problem turned up during the post-experimental interview. However, the subject might just as well have asked a corresponding question during interaction with the system.

Even if an SLDS is able to conduct a perfectly cooperative dialogue, it will need to initiate *repair and clarification meta-communication* whenever it has failed to understand the user, for instance because of speech recognition or language understanding failure. *P14* (SP9 in Table 3) states what the cooperative system should do in case of failure to understand utterances made by the user. Together with the two new specific principles which derive from the user test, SP10 and SP11, we have subsumed SP9 by the new generic principle GP13 (Table 2).

Finally, *P3* (SP3), *P7* (SP6), *P9* (SP7), *P11* (SP1) and *P13* (SP2), are all specific principles subsumed by already established generic principles. SP6, for instance, is subsumed by GP11 (background knowledge). SP6 was developed from examples of user misunderstanding due to reasoning by analogy. For instance, the fact that it is possible to make reservations of stand-by tickets on international flights might lead users to conclude (erroneously) that this is also possible on domestic flights.

5. Testing the Principles

Having consolidated the principles of cooperative system dialogue through comparison with Gricean theory, we were keen to analyse how the principles worked out in the user test of the implemented system. It should be noted, once again, that the principles had not been systematically used as guidelines in designing the implemented system. The user test thus cannot be considered a test of the principles in the crucial sense of indicating to which extent successful design of cooperative (system) dialogue may result from using the principles as design guidelines. Rather, the user test worked as a test of the *scope* of the principles. It provided an indication of whether the present set of principles converge on a complete set. If analysis, along the lines described in Section 3, of user-system interaction in the user test would produce a significant number of novel principles, then the present set of principles are still far from complete. On the other hand, if the analysis failed to produce more principles, then this would at least suggest that convergence on completeness is well under way.

The system was tested with 12 users from the intended user population. The resulting 57 transcribed reservation dialogues were scenario-based and covered the full functionality of the system.

We analysed the dialogues to detect all those deviations from expected user behaviour that would suggest problems of user-system interaction caused by non-cooperative dialogue design [12]. Each problem identified was represented as a violation of a principle of cooperative system dialogue. As it turned out, almost all of the 117 individual problems identified could be ascribed to violations of the cooperative principles in Table 2. No new *generic* principles had to be added. We only had to add two *specific* principles of meta-communication (SP10 and SP11 in Table 2). Since meta-communication had not been simulated during WOZ and the WOZ corpus thus contained very few examples of meta-communication, this came as no surprise. The following principles were found violated at least once: GPs 1, 5-7 and 10-13, SPs 2, 4-6, 8 and 10-11. The following principles were not found violated: GPs 2-4 and 8-9, SPs 1, 3, 7 and 9. In other words, the user test confirmed the broad coverage of the principles with respect to cooperative spoken user-system dialogue. Less flattering, the test clearly demonstrated several deficiencies in our design of cooperative system dialogue.

6. Concluding Discussion

We have described the development, from a relatively large corpus of simulated human-machine spoken dialogue, of a set of principles or experience-based hypotheses of cooperative system dialogue. The principles were shown to include as a sub-set a well-established body of maxims of cooperative human-human dialogue. The principles have a considerably wider scope than the maxims and split into generic and specific principles. At the generic level, the principles address three aspects of cooperative dialogue which are not addressed by the maxims. The specific principles have no counterparts among the maxims. Yet these principles appear useful to SLDS design. What we need in order to discover dialogue problems at an early stage, is to know what to look for in the emerging dialogue structure. The specific principles extend the generic principles by further specifying their import. Analysis of the corpus that was produced from the user test of the implemented system shows that the generic principles are adequate for, that is, able to subsume, all the identified dialogue problems. The user test corpus analysis did, however, increase the number of specific principles by two which both address dialogue issues that were not prominent in the original corpus of simulated human-machine dialogue. Jointly, these results suggest that the principles of cooperative system dialogue represent a step towards a more or less complete and practically applicable set of design guidelines for cooperative SLDS dialogue.

Two further lines of investigation must be pursued in order to test and improve the completeness and practical utility of the principles. First, it cannot be excluded at this stage that the principles are somehow tied to the task domain and dialogue complexity of our particular SLDS. Analysis of dialogue problems caused by systems that address different task domains or have lower or higher dialogue complexity than our system may thus reveal additional specific or even generic principles. Secondly, principles of cooperative dialogue are not necessarily the same as practically applicable design guidelines. As Baber remarked (Section 1), it is not obvious how to apply the Gricean maxims as design guidelines. Or, speaking about principles instead, an SLDS designer who simply receives the principles as represented in Table 2, may not quite know what to do with them in practice. Jointly, these two lines of investigation raise issues such as: how *comprehensible* are the principles as they stand? How *adequate* are they to the development of systems different from our own? Does their use have *measurable effects?* How should the principles be *communicated* to

achieve maximum effect? Based on answers to questions such as these, the task will be to seek to provide the necessary support for the principles to become of maximum benefit to dialogue design practice, thereby reducing the cost of producing habitable dialogue for SLDSs.

Methodologically, using the principles as guidelines means to apply them to analytical "walk-throughs" through the emerging dialogue structure of the SLDS that is being designed (Section 3). This requires training and skill. We believe that a representation of the principles which includes their justification, such as that of Table 1, might be of help. An extensive set of example violations might help as well. Neither of those support functions have been sufficiently provided above. We hope, however, to have provided sufficient information for other SLDS designers to start trying out how the principles work as practical guidelines.

References

1. J. Peckham, "Conversational Interaction: Breaking the Usability Barrier," Proceedings of the ESCA workshop on Spoken Dialogue Systems, Vigsø, Denmark, 1995, pp. 1-8.
2. N. M. Fraser and G. N. Gilbert, "Simulating Speech Systems," Computer Speech and Language 5, 1991, pp. 81-99.
3. C. Baber, "Developing Interactive Speech Technology," In C. Baber and J. M. Noyes (Eds.), Interactive Speech Technology: Human Factors Issues in the Application of Speech Input/Output to Computers, Taylor and Francis, London, 1993, pp. 1-18.
4. P. Grice, "Logic and Conversation," In P. Cole and J. L. Morgan (eds.), Syntax and Semantics, Vol. 3, Speech Acts, Academic Press, New York, 1975, pp. 41-58. Reprinted in P. Grice, Studies in the Way of Words, Harvard University Press, Cambridge MA, 1989.
5. B. Schneidermann, "Designing the User Interface," Addison-Wesley, Reading MA, 1987.
6. H. Dybkjær, L. Dybkjær and N. O. Bernsen, "Design, Formalisation and Evaluation of Spoken Language Dialogue," Proceedings of the TWLT9 Workshop, Enschede, 1995, pp. 67-82.
7. H. Aust, and M. Oerder, "Dialogue Control in Automatic Inquiry Systems," Proceedings of the ESCA Workshop on Spoken Dialogue Systems, Vigsø, Denmark, 1995, pp. 121-124.
8. R. Cole, D. G. Novick, M. Fanty, P. Vermeulen, S. Sutton, D. Burnett, and J. Schalkwyk, "A Prototype Voice-Response Questionnaire for the US Census," Proceedings of the ICSLP '94, Yokohama, 1994, pp. 683-686.
9. E. Bilange, "A Task Independent Oral Dialogue Model," Proceedings of the 5th EACL, Berlin, 1991, pp. 83-88.
10. H. Bourlard, "Towards Increasing Speech Recognition Error Rates," Proceedings of Eurospeech '95, Madrid, 1995, pp. 883-894.
11. N. O. Bernsen, H. Dybkjær and L. Dybkjær, "Cooperativity in Human-Machine and Human-Human Spoken Dialogue," Discourse Processes 21, 1996, pp. 213-236.
12. L. Dybkjær, N.O. Bernsen and H. Dybkjær, "Reducing Miscommunication in Spoken Human-Machine Dialogue," AAAI '96: Workshop on detecting repairing and preventing human-machine miscommunication, Portland, 1996, pp. 29-36.