Designing Co-Operativity in Spoken Human-Machine Dialogue

Laila Dybkjær, Niels Ole Bernsen and Hans Dybkjær
Centre for Cognitive Science, Roskilde University
PO Box 260, DK-4000 Roskilde, Denmark
emails: laila@cog.ruc.dk, nob@cog.ruc.dk, dybkjaer@cog.ruc.dk

Abstract

Dialogue model design for spoken language dialogue systems (SLDSs) is still based mainly on common sense, experience and intuition, and trial and error, rather than on established design principles. Co-operativity in dialogue is crucial to habitable human-machine spoken dialogue. The paper presents a set of principles of co-operative user-system dialogue which have been derived from a corpus of task-oriented spoken human-machine dialogue. The set of principles is shown to include as a sub-set an established body of principles of co-operative human-human dialogue. Analysis of results from a user test of an implemented SLDS prototype shows the set of principles to be adequate to account for the dialogue problems identified in the test corpus. Both empirical and theoretical grounds thus indicate that the principles presented in the paper may constitute a comprehensive set of guidelines for the design of co-operative human-machine dialogue.

1. Introduction

Current task-oriented spoken language dialogue systems (SLDSs) technologies are based on the assumption of co-operative user dialogue behaviour. This fact does not, however, pose much of a problem for dialogue designers because the penalty for non-co-operativity is that users fail to get their task done. There is no point in designing the dialogue for non-cooperative users who want to make the system fail. Indeed, this design goal is impossible to achieve. However, habitable user-system dialogue requires that also the *system's* dialogue behaviour be co-operative. 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 on the dialogue itself rather than on the task domain of the dialogue, and is typically initiated for purposes of clarification and repair. In particular the speech recognition capabilities of current

SLDSs are still fragile [8]. Sophisticated meta-communication functionality is needed to overcome the effects of system misrecognitions [17]. Thus, to the extent possible, the user's needs to initiate clarification and repair meta-communication should not be compounded by non-cooperative system behaviour. At any stage during dialogue, the co-operative user should know what to do and how to do it, without having been led astray by a non-cooperative system. A crucial dialogue design goal, therefore, is to optimise system dialogue co-operativity in order to prevent user-initiated clarification and repair meta-communication. Such meta-communication tends to increase to a level beyond what is currently technically feasible, the demands on the system's language comprehension and dialogue management capabilities and decrease the user's satisfaction in communicating with the system. The practical problem therefore becomes: how do dialogue designers design co-operative system dialogue behaviour? To our knowledge, whereas there is agreement in the literature that current, task-oriented SLDSs require co-operative user dialogue behaviour [7, 13], the question of how to design co-operative system dialogue has not been addressed in any systematic way. There is a clear need to do so, particularly if the result would be a set of guidelines for co-operative system dialogue design for effective and systematic use as development and evaluation tools during early design. This might significantly reduce development time by reducing the need for lengthy Wizard of Oz experimentation, controlled user testing, and field trial cycles, thereby reducing overall development costs.

In the course of developing, implementing and testing an SLDS prototype in the Danish Dialogue project, we have developed a set of principles of co-operative system dialogue. Given the way these principles were developed, compared to well-established theoretical results from the analysis of co-operativity in human-human dialogue, and subsequently tested in a user test of the implemented system, we believe that the principles deserve consideration by the SLDS dialogue design and evaluation community. It seems likely that the principles cover most, if not all, aspects of co-operative system dialogue design and hence might be useful to the design and evaluation of the many SLDSs which are now making their way from research laboratories through field testing to product development. The most advanced among these systems have system-directed dialogue which means that system co-operativity is a main design goal.

The Danish SLDS prototype addresses the domain of domestic flight ticket reservation and has been developed in collaboration with the Center for Person Kommunikation at Aalborg University and the Centre for Language Technology in 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. It 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 with pre-recorded speech. The system is a representative example of advanced state-of-the-art systems. Comparable SLDSs are found in [1, 9, 14].

In what follows, Section 2 provides an account of the development of our principles of co-operative system dialogue leading to an expression of the principles themselves. The principles were derived from a corpus of simulated human-machine dialogue which was recorded during the design of a dialogue model for the Danish dialogue system. The purpose of the principles is to prevent users from having to initiate clarification and repair meta-communication because of non-cooperative dialogue design. Section 3 compares the principles with Grice's maxims of co-operativity in human-human dialogue. We had developed our principles independently of Grice's work and only subsequently became aware of the close relationship between that work and our own efforts. It turned out that Grice's maxims could be mapped onto a sub-set of our principles of co-operativity, which suggested that our efforts at principle development were on the right track. The theoretical efforts of articulating the principles and comparing them with Grice's maxims took place in parallel with the implementation of the Danish dialogue system and after the Wizard of Oz experiments preceding implementation. Thus, the principles were *not* used as design guidelines during implementation. This meant that the test, with naive users, of the implemented system could be considered a test of the completeness of the principles. Section 4 describes the results of that test. The results indicate that the application, during early dialogue design, of co-operative dialogue design principles can help SLDSs designers prevent user-initiated clarification and repair meta-communication and thereby increase the habitability of their products. Section 5 concludes and discusses how the principles of co-operative system dialogue behaviour may be developed into low-cost guidelines for use in SLDS design practice as well as in SLDS evaluation.

2. Developing Principles of Co-Operative System Dialogue

Dialogue design for SLDSs consists in defining and refining a set of design requirements or constraints which are traded off against one another in an iterative development process until an acceptable result has been achieved. No matter what methods are used during this phase, the central point of dialogue development is to observe and analyse the user-system interaction to assess whether the dialogue model satisfies the design requirements and is adequate in terms of functionality and usability. User and system dialogue problems should be identified and analysed, and results used to change the dialogue model before performing a new iteration of observation and analysis.

2.1 The Wizard of Oz Experiments

Dialogue models for SLDSs are often designed by using the Wizard of Oz method (WOZ). WOZ is an iterative simulation technique which is well suited for the development and testing of dialogue models prior to implementation. During each iteration a human (the ‘wizard’) simulates the system or parts thereof in dialogue with users who should be made to believe that they are speaking to a real system [15]. The dialogues are recorded, transcribed and analysed, and the results used to improve the dialogue model. This iterative process continues until a dialogue model has been achieved that satisfies the design requirements. The model is then implemented and tested on representative samples of the intended user population. The advantage of using the WOZ method is that user and system problems can be removed prior to the implementation of the dialogue model. Given the state of current SLDS development environments, the extra cost of performing WOZ experiments will often be less than the cost of making changes to the implemented system in the light of results of controlled user testing or field testing of the system. However, even if the WOZ method is being used, it remains true that 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. This means that if, during WOZ, the dialogue designers are not very careful in addition to being lucky, many user and system problems may still remain to be discovered during implementation and subsequent tests of the system.

Seven WOZ iterations were performed to produce the dialogue model for the Danish dialogue system [12]. Since the application is accessed over the telephone, real-time performance was considered a constraint which had to be satisfied by a usable system. In the context of the chosen hardware and software including the speech recogniser, the real-time constraint gave rise to additional constraints:

- At most 100 words can be active in memory at a time to enable real-time performance.
- The average user utterance length should not exceed 3-4 words.
- The maximum user utterance length should not exceed 10 words.

The two last-mentioned constraints served the additional purpose of maintaining the recogniser error rate at an acceptable level. Furthermore, because of limited project resources the system vocabulary size was set to about 500 words.

Apart from real-time performance, the main usability constraints were: sufficient task domain coverage, robustness, natural forms of language and dialogue, and dialogue flexibility. These usability constraints had to be traded off against the above-mentioned resource constraints and technological constraints. It was the task of the WOZ experiments to optimise the trade-offs [10].

The first five WOZ iterations served to train the wizard and produce an outline dialogue model. Each iteration generated only a few dialogues. The dialogue model was initially represented as a loosely ordered set of predefined phrases but was soon turned into a graph structure (a state transition network) in order to facilitate the wizard's job (cf. Figure 1). The graph has predefined system phrases in the nodes and expected user input contents along the edges. Users (subjects) were exclusively system designers and colleagues. The last two WOZ iterations were considerably larger than the first ones and aimed at defining the dialogue model to be implemented. Each iteration involved 12 subjects mostly from outside the lab. None of the (9) external subjects had tried the system prior to the WOZ experiment. External subjects were selected so that half of them had a background as secretaries and the other half were computer scientists. The expected end-user group is mainly secretaries. The computer scientists were included in order to study the reactions of people who had general system knowledge.

Throughout the experiments, interaction with the system was based on scenarios, i.e. domain relevant tasks. The first four WOZ iterations were based on a set of ten scenarios which were simply considered a set of cases for which the system should work and which were mainly used for domain and task exploration. Most decisions on precise reservation details such as date of departure were left to the subjects. In the last three WOZ iterations a new set of 28 scenarios was used. The scenarios were designed on the basis of the dialogue structure that emerged from the fourth WOZ iteration. By then the scenarios could be designed in a more systematic way, as most of the domain and task structure had been uncovered.

Each subject in the fifth, sixth and seventh iterations received (i) a letter which briefly introduced the system and informed on the experiment, (ii) four scenarios and (iii) a questionnaire to be filled in and returned immediately after the subject's interaction with the system. Subjects were not told in advance that the system was simulated. In a debriefing telephone interview after the session subjects were asked in WOZ7 whether they believed that they had interacted with a real system. The majority of external subjects believed that the system was real. Each of the two last WOZ iterations produced a corpus of 47 dialogues. From the seven iterations a total of 125 dialogues were transcribed amounting to about seven hours of spoken language dialogue. 24 different subjects were involved in the seven iterations.

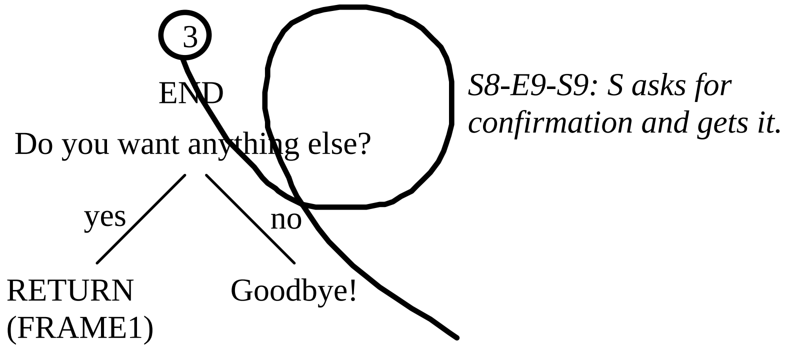
After each iteration the transcribed dialogues were analysed and evaluated with respect to the extent to which the design constraints had been satisfied. Evaluation results were used as a basis for improving the dialogue model before the next WOZ iteration. In the first iterations it was easy to find suggestions for improvement by merely listening to the dialogues or looking through the transcriptions. However, as the dialogue model improved, more sophisticated and systematic methods of dialogue analysis became necessary. We began to match the scenarios to be used in the following iteration against the current dialogue graph structure in order to discover and, as far as possible, 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 occurred during user-system dialogue. Many problems were discovered analytically through the scenario-based walkthroughs of the dialogue model.

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 assess improvements in system co-operativity and discover actual user problems. The dialogue model representation was split into a number of sub-graphs corresponding to different sub-tasks. Each transcribed dialogue was plotted onto the dialogue sub-graphs. Deviations from the sub-graphs indicated unexpected user or system behaviour. The deviations were marked and the reason(s) for the deviations analysed. This plotting of the transcribed dialogues onto the dialogue structure is very similar to the scenario-based walkthroughs but aims at discovering actual user problems. 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 the confirmation, the subject asked for it.

The following dialogue fragment provides the background for the subject's deviation from the dialogue model. The subject has made a change to a flight reservation. After the user has indicated the change, the conversation continues:

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 onwards 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 on change of reservation.



S8-E9-S9: S asks for confirmation and gets it.

Figure 1: A plotted END sub-graph from WOZ6. The boldfaced loop deviating from the graph path shows unexpected user dialogue behaviour which may reveal a dialogue design problem. The encircled number (3) refers to the point in the previously traversed graph from which the subject jumped to the END sub-graph. The deviation is annotated with numbered reference (in italics) to the relevant transcribed utterances and a description of the deviation. E refers to the experimenter and S to the subject.

2.2 Developing Design Principles for Co-Operative System Dialogue

At the end of the WOZ design phase, we began a more theoretical, forward-looking exercise of categorising identified dialogue design problems and expressing the corresponding dialogue design principles. To this end, we plotted all the transcribed user-system dialogues from WOZ3 onwards onto their corresponding graphs. In addition, we compared each dialogue model graph pair (WOZn and WOZn+1) in order to identify and analyse all changes made to the dialogue model from WOZ3 through to WOZ7. To illustrate the latter process, Figures 2 and 3 allow comparison of part of the TIME sub-graphs in WOZ5 and in WOZ6, respectively. Some of the main differences between the two sub-graphs are: WOZ5 does not include the discount option, which reveals a flaw in task domain coverage. The ‘fully booked’ message does exist in WOZ5 but is represented in a separate sub-graph. When following the 'wrong time' and 'time' edges of the WOZ5 sub-graph, users are never allowed to state a precise hour of departure. Having provided information on the closest departure times, the system will go on to address a new topic. In doing so the system's dialogue contribution is not relevant, i.e. is not appropriate to the immediate needs at this stage of the transaction. This problem has been repaired in WOZ6.

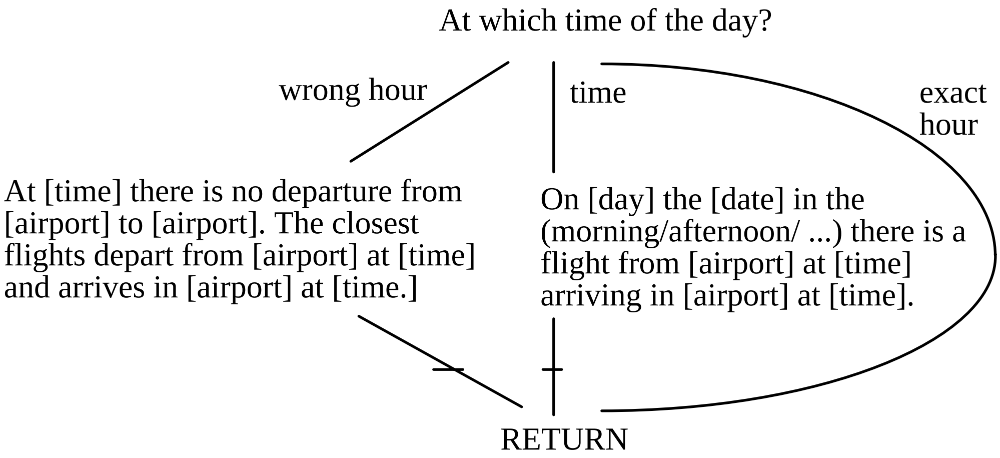

At which time of the day?
wrong hour
time
exact hour
At [time] there is no departure from [airport] to [airport]. The closest flights depart from [airport] at [time] and arrives in [airport] at [time.]
On [day] the [date] in the (morning/afternoon/ ...) there is a flight from [airport] at [time] arriving in [airport] at [time].
RETURN

Figure 2: Part of the TIME sub-graph from WOZ5.

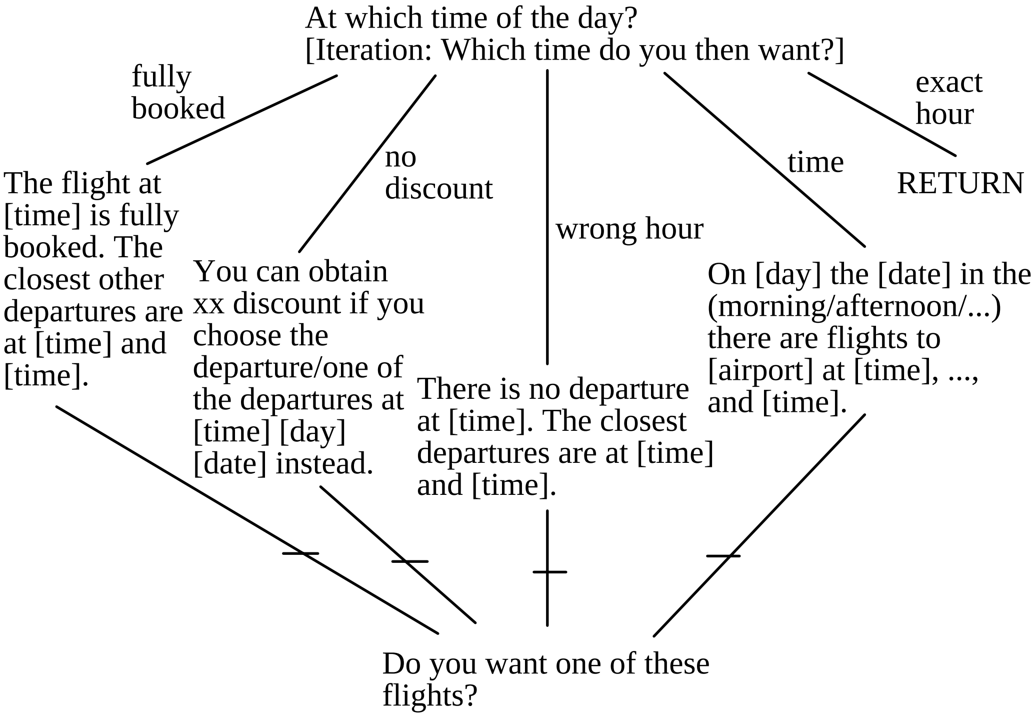

At which time of the day?
[Iteration: Which time do you then want?]
fully booked
no discount
wrong hour
time
exact hour
RETURN
The flight at [time] is fully booked. The closest other departures are at [time] and [time].
You can obtain xx discount if you choose the departure/one of the departures at [time] [day] [date] instead.
There is no departure at [time]. The closest departures are at [time] and [time].
On [day] the [date] in the (morning/afternoon/...) there are flights to [airport] at [time], ..., and [time].
Do you want one of these flights?

Figure 3: Part of the TIME sub-graph from WOZ6.

Based on the material illustrated above, the actual and potential user problems identified in the WOZ experiments were analysed, classified and represented as violations, made by the dialogue system, of principles of co-operative dialogue. Each problem was considered a case in which the system in addressing the user had violated a principle of co-operative dialogue. The principles of co-operative dialogue were made explicit, based on the problems analysis. In addition, we analysed how the system's utterances had been, or sometimes should be, improved to minimise user-initiated clarification and repair meta-communication [2, 6]. To illustrate the WOZ corpus analysis, we present below an example of an identified user problem type (a) and the co-operative principle (termed 'design commitment') which has been violated (b). A justification of the principle is provided (c), followed by examples of how the principle was found to be violated (d). Under (d) we note whether a particular example was discovered empirically (i.e. from actual user problems) or analytically (i.e. through design analysis revealing a potential user problem). Finally, a solution to each particular problem is proposed and sometimes discussed (e). This template (a-e) was applied to each problem that was identified [2]. In the original report on the data [2], the principles were illustrated by 36 concrete examples of their violation, but the total number of examples in the corpus were +100.

(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. This 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. However, there were other situations in which shortcuts would have been desirable as well. The need for shortcuts perhaps could be met by allowing the use of keywords at certain points in the dialogue. This might work with expert users. For non-expert users, however, large numbers of keywords represent a non-optimal solution and would probably require access to a written system manual.

2.3 Design Principles for Co-Operative System Dialogue

The WOZ corpus analysis led to the identification of 14 principles of co-operative human-machine dialogue (see Table 1). The table includes a justification of each principle, which serves the additional purpose of clarifying the meaning and scope of the principle. 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 repair or clarification meta-communication, because this is the strategy naturally adopted in human-human conversation in such cases.

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.

Principles (continued)	**Justification** (continued)
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).	Need for non-superfluous interaction with the system.
P11. Be fully explicit in communicating to users the commitments they have made.	Users need feedback from the system on the commitments made.
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 co-operative SLDS dialogue design principles (left-hand column) and their justifications (right-hand column).

3. Confirming the Principles of Co-Operative System Dialogue

The work described in the previous section led to the development of general principles of co-operative human-machine dialogue. Most of the 14 principles aimed at improving system co-operativity. Only two principles (P1 and P8, see Table 1) were aimed at improving user co-operativity. Having developed these principles we became aware of a link between our work and Grice's Co-operative Principle and maxims [16]. Grice's Co-operative Principle (CP) says that, to act co-operatively 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". Grice proposes that the CP can be explicated in terms of four groups of simple maxims which are not claimed to be jointly exhaustive nor to have been generated on a principled theoretical basis other than the CP itself. The maxims are shown in Table 2.

Dialogue aspect	**M No.**	**Maxim**	**Grice's comments**
Group 1: **Quantity**	M1	Make your contribution as informative as is required (for the current purposes of the exchange).	Grice observes that M2 is closely related to M5. In other words, maxims, as stated by Grice, are not mutually exclusive but may overlap.
	M2	Do not make your contribution more informative than is required.	
Group 2: **Quality**	M3	Do not say what you believe to be false.	Grice notes that M3 and M4 seem presupposed by the other maxims. He nevertheless refrains from putting them in a different category from the rest.
	M4	Do not say that for which you lack adequate evidence.	
Group 3: **Relation**	M5	Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.	Grice points out that the concept of relevance is in need of further explication (see, e.g. [18]).
Group 4: **Manner**	M6	Avoid obscurity of expression.	Grice notes that there may well be more maxims in Group 4.
	M7	Avoid ambiguity.	
	M8	Be brief (avoid unnecessary prolixity).	
	M9	Be orderly.	

Table 2: Grice’s maxims and comments [16]. The left-hand column presents a higher-level grouping of the maxims proposed by Grice. We view the groups as addressing different aspects of dialogue.

Grice focuses on dialogues in which the interlocutors want to achieve a shared goal. In such dialogues, he claims, adherence to the CP and the maxims is rational because this ensures that the interlocutors pursue the shared goal most efficiently. Task-oriented dialogue, such as that for which our SLDS has been designed, would seem to be a prototypical case of shared-goal dialogue. However, Grice did not develop the maxims for the specific purposes of preventing communication failure and avoiding interlocutor-initiated clarification and repair meta-communication in shared-goal dialogue. Rather, his interest lies in the inferences which an interlocutor is able to make when the speaker deliberately does not adhere to one of the maxims. He calls such deliberate messages ‘conversational implicatures’. In SLDS design we are obviously not interested in including such messages in the system’s utterances. Grice’s maxims, although having been conceived with a different purpose in mind, nevertheless serve the same objective as do our principles, namely that of preventing interlocutor-initiated clarification and repair meta-communication. It is exactly when a human or, for that matter, an SLDS, *non-deliberately* fails to adhere to a maxim, that the interlocutor is likely to initiate repair or clarification meta-communication. Thus, the main difference between Grice’s work and ours seems to be that the maxims were developed to account for co-operativity in human-human dialogue, whereas our principles were developed to account for co-operativity in human-machine dialogue.

3.1 Principles which are Reducible to Maxims

Having discovered the link between our principles and Grice’s maxims we made a detailed analysis of the relationship between principles and maxims [4]. At least superficially, our set of principles is considerably larger than Grice's set of maxims. The analysis demonstrates that a sub-set of our principles can be reduced to, and replaced by, the maxims. Briefly, referring to Tables 1 and 2 above, P5 may be replaced by M6 and M7, P6 by M1 and M9, P10 by M2 and M5, and P12 by M8 [4]. These maxims are capable of performing the same job as do the corresponding principles, in guiding the design of co-operative human-machine dialogue. In fact, the maxims perform the better job in view of the facts that (i) M6 and M7 spell out the intended contents of the infelicitously expressed P5, and (ii) M1 and M9 replace P6. The only maxims which have no corresponding principles are the maxims of quality M3 and M4. The reason is that one does not design an SLDS which provides false or unfounded information to users. 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. Truth and evidence form a major concern during system implementation, as it cannot be allowed, for example, that the system confirms information which has not been checked with the database and which might be false or impossible. Grice observes that the maxims of quality in general, and M3 in particular, have the special status of being presupposed by the rest of the maxims.

Another result of analysing the relationship between principles and maxims is the distinction between *generic* and *specific* principles. Grice's maxims are all generic. A generic principle may subsume one or more specific principles which specialise the generic principle such as to deal with certain classes of situations. Specific principles are important in SLDS design. The following three principles are specific and can be subsumed by one of Grice’s maxims:

> P3. Provide same formulation of the same question (or address) to users everywhere in the system’s dialogue turns.

P3 represents a precaution against the occurrence of ambiguity in machine speech and can be viewed as a specific principle subsumed by M7 (ambiguity).

> P11. Be fully explicit in communicating to users the commitments they have made.
> P13. Provide feedback on each piece of information provided by the user.

P11 and P13 are closely related, specific principles. Feedback is a special type of co-operative dialogue contribution in which the speaker explicitly expresses an interpretation of the interlocutor's previous dialogue contribution(s). We propose that P11 and P13 are subsumed by M1 (informativeness).

The fact that a sub-set of our principles of co-operative human-machine dialogue is near-equivalent to the Gricean maxims suggests that Grice's maxims are valid not only for shared-goal human-human dialogue but also for human-machine dialogue.

3.2 Principles which are not Reducible to Maxims

The remaining principles appear irreducible to maxims. Of these principles some are generic whereas others are specific. Moreover, the new generic principles express three new dialogue aspects in addition to the four aspects identified by Grice, i.e. 'quantity', 'quality', 'relation' and 'manner' (cf. Table 2). The new aspects are: dialogue partner asymmetry, background knowledge, and repair and clarification.

Dialogue partner asymmetry exists, roughly, when one or more of the dialogue partners is not in a normal condition or situation, such as having impaired hearing or being located in a particularly noisy environment. The non-normal dialogue partner should inform the dialogue partner(s) about the particular non-normal characteristics which they should take into account in order to behave co-operatively. In such cases, dialogue co-operativity depends on the interlocutor(s) taking into account the non-normal participant's special characteristics. Since, obviously, SLDSs are non-normal dialogue partners, their designers should make users aware of their non-normal characteristics if clarification and repair meta-communication is to be avoided. The following two principles address partner asymmetry.

> P1. Provide clear and comprehensible communication of what the system can and cannot do.
> P8. Provide clear and sufficient instructions to users on how to interact with the system.

Since our SLDS has limited task capabilities and is intended for walk-up-and-use application, it must provide users with an up-front mental model of what it can and cannot do, as expressed in P1. P8 has an analogous role. P1 and P8 introduce two new properties of dialogue co-operativity, namely partner asymmetry and speaker's obligation to inform the interlocutor(s) of any non-normal speaker characteristics. P1 and P8, therefore, cannot be subsumed under any other principle or maxim. We propose a new generic principle (P15-NEW) which subsumes P1 and P8.

> P15-NEW. Inform the dialogue partners of important non-normal characteristics which they should take into account in order to behave co-operatively in dialogue.

Background knowledge and differences in background knowledge is an important aspect of dialogue. Interlocutors have different background knowledge. Such differences often have to be taken into account in order to maintain co-operative dialogue. Human speakers either have built in advance, or adaptively build during dialogue, a model of the interlocutor which serves to guide co-operative dialogue behaviour. Increased user adaptivity in this sense is an important goal in SLDS design [5, 11].

> P4. Take users' relevant background knowledge into account.

P4 cannot be reduced to M1 (informativeness), since M1 does not include the notions of background knowledge and differences in background knowledge among interlocutors. Moreover, a speaker may adhere perfectly to 'exchange purpose' while ignoring important elements of the interlocutor's background knowledge. For similar reasons, M5 (relevance) cannot replace P4. In fact, P4 appears to be presupposed by maxims M1, M2 and M5 to M9 in the sense that it is not possible to adhere to any of these maxims without adhering to P4.

P7 and P9 are two specific principles which may both be subsumed by P4.

> P7. Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.

> P9. Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).

In their proper domains, SLDSs should behave as experts towards their users. They should therefore have sufficient task domain knowledge as stated in P2.

> P2. Provide sufficient task domain coverage.

P2 is a specific principle. However, because it deals with speaker's knowledge, it cannot be subsumed under P4 above. We propose to introduce a new generic principle which mirrors P4 and subsumes P2:

> P16-NEW. Take into account legitimate partner expectations as to your own background knowledge.

Even if an SLDS is able to conduct a perfectly co-operative dialogue, it will need to initiate *repair and clarification meta-communication* whenever it has failed to understand its human user, for instance because of speech recognition or language understanding failure:

> P14. Provide ability to initiate repair if system understanding has failed.

P14 states what the co-operative speaker should do in case of communication failure. P14 is a generic principle and cannot be subsumed under M1 (informativeness) which does not address issues of meta-communication. P14 may be replaced by the slightly revised P14*:

> P14*. Initiate repair or clarification meta-communication in case of communication failure.

3.3 The Final Set of Principles

It may be concluded that there are more principles of co-operativity in human-machine dialogue than those identified by Grice. Three groups of principles reveal aspects of co-operative dialogue which were not addressed by the maxims. This yields a total of seven dialogue aspects, each of which is addressed by one or more generic principles (see Table 3). Some of the generic principles subsume one or more specific principles (see Table 4). Specific principles SP10 and SP11 in Table 4 were developed as a result of the user test of the Danish SLDS (see Section 4).

4. Testing the Principles of Co-Operative System Dialogue

A user test of the implemented system was carried out with a simulated speech recogniser [3]. The recognition accuracy would be 100% as long as users expressed themselves in accordance with the vocabulary and grammars known to the system. Otherwise, the simulated recogniser would turn the user input into a string which only contained words and grammatical constructions from the recogniser's vocabulary and rules of grammar. The test was carried out in a way similar to the two last WOZ experiments (cf. Section 2). It involved 12 external subjects who had never tried the system. Each subject received four scenarios. Subjects conducted the dialogues over the telephone in their normal work environments in order to make the situation as realistic as possible. Each dialogue between a subject and the dialogue system was recorded. All transactions between the individual system modules were logged.

A total of 57 dialogues were recorded. Some subjects repeated a task if they failed to achieve their goals in the first dialogue attempt. The recorded dialogues were transcribed and analysed. In order to test our principles of co-operative dialogue design and obtain a detailed overview of user and system problems in the user test, we identified all such problems in the transcribed corpus. The dialogue being system-directed, we could specify the system's questions in a fixed tabular format. For each scenario and system question we then specified the key contents of the expected user answer. This provided a normative model of the completion of each scenario used in the test. After transcription of the test corpus, the key contents of the actual user answers were added to a table representing the relevant scenario. Each deviation from the expected user input indicated a potential problem and was carefully analysed. This analysis often required use of the transcribed dialogue itself as well as the logged transactions between the system modules during the dialogue.

Dialogue aspect	**GP No.**	**Generic principle**
Group 1: **Informativeness**	GP1	*Make your contribution as informative as is required (for the current purposes of the exchange).
	GP2	*Do not make your contribution more informative than is required.
Group 2: **Truth and evidence**	GP3	*Do not say what you believe to be false.
	GP4	*Do not say that for which you lack adequate evidence.
Group 3: **Relevance**	GP5	*Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.
Group 4: **Manner**	GP6	*Avoid obscurity of expression.
	GP7	*Avoid ambiguity.
	GP8	*Be brief (avoid unnecessary prolixity).
	GP9	*Be orderly.
Group 5: **Partner asymmetry**	GP10	Inform the dialogue partners of important non-normal characteristics which they should take into account in order to behave co-operatively in dialogue.
Group 6: **Background knowledge**	GP11	Take partners' relevant background knowledge into account.
	GP12	Take into account legitimate partner expectations as to your own background knowledge.
Group 7: **Repair and clarification**	GP13	Initiate repair or clarification meta-communication in case of communication failure.

Table 3: Generic principles of co-operative spoken dialogue. Generic principles are expressed at the same level of abstraction as are the Gricean maxims (marked with an *). The left-hand column characterises the aspect of dialogue addressed by each principle. Comparison with Table 2 shows that the three first dialogue aspects have been re-named.

Table 5 shows a scenario table in which two problems were identified: (i) undersupported user navigation and (ii) answering several questions at a time. The user wants a roundtrip ticket. In the tested version of our SLDS, roundtrip tickets can only be booked as two one-way tickets. Not having been informed about this, the user attempts to book a return ticket. The first problem (i) lies in the lack of information to users on how to navigate the system in order to book roundtrip tickets. The second problem (ii) probably occurs because the user found no other way of telling the system that the departure airport for the home journey is different from the destination of the out journey. In the system's opening instructions to users, these are told to answer the system's questions one at a time.

After a first iteration of describing the identified problems, these were seen to belong to one of two broad types, i.e. *system problems* and *user errors.* System problems demonstrate flaws in, i.a., the design of the system's language processing or dialogue design components. User errors were of many types, ranging from misreading of the scenarios and initiating repair through questions which the system was unable to understand, through to non-co-operative dialogue behaviour. A detailed analysis is in preparation. We shall focus on the dialogue design problems in what follows. For each instance of a dialogue design problem type we made a diagnosis and proposed a cure. Figure 4 shows an example in which the user has asked both to depart at 7.20 (am) and to have discount. The diagnosis shows that the system imposes an unjustified priority among these two goals. The cure proposes how to revise the system's handling of inconsistent user input.

Dialogue aspect	**GP No.**	**SP No.**	**Specific principle**
Group 1: **Informativeness**	GP1	SP1	Be fully explicit in communicating to users the commitments they have made.
	GP1	SP2	Provide feedback on each piece of information provided by the user.
Group 2: **Truth and evidence**			
Group 3: **Relevance**			
Group 4: **Manner**	GP7	SP3	Provide same formulation of the same question (or address) to users everywhere in the system’s dialogue turns.
Group 5: **Partner asymmetry**	GP10	SP4	Provide clear and comprehensible communication of what the system can and cannot do.
	GP10	SP5	Provide clear and sufficient instructions to users on how to interact with the system.
Group 6: **Background knowledge**	GP11	SP6	Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.
	GP11	SP7	Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).
	GP12	SP8	Provide sufficient task domain coverage.
Group 7: **Repair and clarification**	GP13	SP9	Provide ability to initiate repair if system understanding has failed.
	GP13	SP10	Initiate clarification meta-communication in case of inconsistent input.
	GP13	SP11	Initiate clarification meta-communication in case of ambiguous user input.

Table 4: Specific principles of co-operative spoken dialogue. Each specific principle is subsumed under a generic principle.

<table>
<tr><td colspan="4">Scenario: G-5-4-a-1 User: 10 Date: January 25 1995</td></tr>
<tr><th>System questions</th><th>Normative user answers</th><th>Actual user answers</th><th>Problems</th></tr>
<tr><td>System already known</td><td>no / yes / -</td><td>-</td><td></td></tr>
<tr><td>Customer number</td><td>2</td><td>yes (2)</td><td></td></tr>
<tr><td>Number of travellers</td><td>1</td><td>1</td><td></td></tr>
<tr><td>ID-numbers</td><td>4</td><td>4</td><td></td></tr>
<tr><td>Departure airport</td><td>Copenhagen</td><td>Copenhagen</td><td></td></tr>
<tr><td>Arrival airport</td><td>Karup</td><td>Karup</td><td></td></tr>
<tr><td>Return journey</td><td>no</td><td>yes</td><td>Under-supported user navigation</td></tr>
<tr><td>Interested in discount</td><td>-</td><td>no</td><td></td></tr>
<tr><td>Day of departure (out)</td><td>January 31</td><td>January 31</td><td></td></tr>
<tr><td>Hour of departure (out)</td><td>around 7:30 / around 19:30</td><td>16:50</td><td></td></tr>
<tr><td>Day of departure (home)</td><td>-</td><td>February 1</td><td></td></tr>
<tr><td>Hour of departure (home)</td><td>-</td><td>18:10 (no departure)
no [does not want one from list]
15:45 from Esbjerg (no departure [from Karup])
yes [wants one from list]
16:20</td><td>Answering several questions at a time</td></tr>
<tr><td>Delivery</td><td>airport / send</td><td>send</td><td></td></tr>
<tr><td>More</td><td>yes</td><td>no</td><td></td></tr>
</table>

Table 5: The table shows the system's questions, expected key contents of user answers, actual key contents of user answers, and problems identified in a subject's completion of Scenario G-5-4-a-1. Contents in brackets (third column) indicate key contents of the system's next utterance. Comments in square brackets are explanatory. One system problem type (undersupported user navigation) and one user error type (answering several questions at a time) were identified.

Table 6 shows the identified dialogue design problem types. Further analysis showed that each problem corresponded to the violation of one or more principles of co-operative system dialogue design. Table 6 shows that no new generic principles of co-operative system dialogue design were found in the analysis of the user test corpus. However, two new specific principles, SP10 and SP11, were found which both address the issue of meta-communication (see Table 4). This is not surprising. System misunderstandings were not simulated in the WOZ experiments which is why the WOZ corpus contains very few examples of meta-communication. As a result, there was little material from which to develop the specific principles of repair and clarification presented in Table 4.

S: U: red discount + out departure time at 7:20; S: no departure at 7:20. However 7:20 does exist, but without discount.
D: S gives priority to discount over time without reason.
C: S should ask U about priority: 7:20 is not a discount departure. Red discount can be obtained on the departures at x, y and z. Which departure do you want. [If U provides a new departure time: S: Do you still want discount? If U: No; S: List non-discount departures].

Figure 4: Example of the dialogue design problem: inconsistent user input. **S** (boldface) means symptom, **D** means diagnosis and **C** means cure. S (normal) means system and U means user.

Dialogue design problem type	**Principle(s) violated**
Ambiguous user input	SP11
Inference problem	SP8
Inconsistent user input	SP10
Insufficient instructions to users about the use of ‘correct’	SP5
Misleading system utterance	GP1 and GP6
Undersupported user navigation	SP5
Incomplete, grammatically incorrect, or irrelevant response	GP1, GP6, and GP5
Missing feedback	SP2
Ambiguous system output	GP7
Database error	GP3

Table 6: Dialogue design problem types identified during the user test. The right-hand column shows the principles of co-operative dialogue design whose violation produced the problems.

5. Conclusion

We have described how a set of principles of co-operative system dialogue were developed from a relatively large corpus of simulated human-machine spoken dialogue. The principles were then shown to include as a sub-set a well-established body of maxims of co-operative human-human dialogue. Moreover, the set of principles has a considerably wider scope than that of the body of maxims. Thus, at the generic level, the principles address three aspects of co-operative dialogue which are not covered by the maxims. In addition, a sub-set of the principles are specific rather than generic principles. The specific principles have no counterparts among the maxims. Analysis of the dialogue corpus that was produced from the user test of the implemented system has shown that the set of generic principles is adequate for, that is, able to subsume, the identified dialogue problems. The corpus analysis did, however, increase the number of specific principles by two principles which both address dialogue issues that were not prominent in the original corpus of simulated human-machine spoken dialogue. These results suggest, we believe, that the principles of co-operative system dialogue discussed above represent a step towards a more or less complete and practically applicable set of guidelines for the design of co-operative SLDS dialogue.

At least two further steps are needed in order to turn the principles into a set of well-tested guidelines for the design of co-operative SLDS dialogue. The first step is to investigate how the principles actually work as guidelines in dialogue design: how comprehensible are they to SLDS designers? How adequate are they for the development of systems different from out own? How can they be used to reveal potential user problems during early design? Does their use have measurable effects? How should the principles be "packaged" to achieve maximum effect? Based on the answers to questions such as these on comprehensibility, adequacy, methodology, effectiveness and communication, respectively, the second step will be to attempt to provide the necessary support for the principles to become of maximum benefit to dialogue design practice.

References

1. Aust, H. and Oerder, M. "Dialogue control in automatic inquiry systems", Proceedings of the ESCA Workshop on Spoken Dialogue Systems, Vigsø, 30 May to 2 June, pp.121-124, 1995.
2. Bernsen, N.O. "Types of user problems in design. A study of knowledge acquisition using the Wizard of Oz", Esprit Basic Research project AMODEUS-2 Working Paper RP2-UM-WP14. In Deliverable D2: Extending the User Modelling Techniques. June, 1993.
3. Bernsen, N.O., Dybkjær, H. and Dybkjær, L. "Exploring the limits of system-directed dialogue. Dialogue evaluation of the Danish dialogue system", Proceedings of Eurospeech ‘95, Madrid, September, pp.1457-60, 1995.
4. Bernsen, N.O., Dybkjær, H. and Dybkjær, L. "Co-operativity in human-machine and human-human spoken dialogue", To appear in Discourse Processes, 1996.
5. Bernsen, N.O., Dybkjær, L. and Dybkjær, H. "A dedicated task-oriented dialogue theory in support of spoken language dialogue systems design", Proceedings of ICSLP ‘94, Yokohama, September, pp.875-878, 1994.
6. Bernsen, N.O., Dybkjær, L. and Dybkjær, H. "Task-oriented spoken human-computer dialogue", Report 6a, Spoken Language Dialogue Systems, Roskilde University, February, 1994.

7. Bilange, E. "A task independent oral dialogue model", Proceedings of the 5th EACL, Berlin, April, pp.83-88, 1991.
8. Bourlard, H. "Towards increasing speech recognition error rates", Proceedings of Eurospeech ‘95, Madrid, September, pp.883-894, 1995.
9. Cole, R., Novick, D.G., Fanty, M., Vermeulen, P., Sutton, S., Burnett, D. and Schalkwyk, J. "A prototype voice-response questionnaire for the US Census", Proceedings of the ICSLP ‘94, Yokohama, September, pp.683-686, 1994.
10. Dybkjær, H., Bernsen, N.O. and Dybkjær, L. "Wizard-of-Oz and the trade-off between naturalness and recogniser constraints", Proceedings of Eurospeech ‘93, Berlin, September, pp.947-950, 1993.
11. Dybkjær, L., Bernsen, N.O. and Dybkjær, H. "Different spoken language dialogues for different tasks. A task-oriented dialogue theory", Human Comfort and Security, Springer Research Report, Springer Verlag, 1995.
12. Dybkjær, L. and Dybkjær, H. "Wizard of Oz experiments in the development of a dialogue model for P1", Report 3, Spoken Language Dialogue Systems, Roskilde University, February, 1993.
13. Eckert, W. and McGlashan, S. "Managing spoken dialogues for information services", Proceedings of Eurospeech ‘93, Berlin, September, pp.1653-1656, 1993.
14. Eckert, W., Nöth, E., Niemann, H. and Schukat-Talamazzini, E. "Real users behave weird - Experiences made collecting large human-machine-dialog corpora", Proceedings of the ESCA Workshop on Spoken Dialogue Systems, Vigsø, 30 May to 2 June, pp.193-196, 1995.
15. Fraser, N.M. and Gilbert, G.N. "Simulating speech systems", Computer Speech and Language 5, pp.81-99, 1991.
16. Grice, P. "Logic and conversation", In Syntax and Semantics, (eds.) Cole, P. and Morgan J.L., Vol. 3, Speech Acts, pp.41-58, New York, Academic Press, 1975. Reprinted in Grice, P. Studies in the Way of Words. Cambridge, MA: Harvard University Press, 1989.
17. Heisterkamp, P. "Ambiguity and uncertainty in spoken dialogue", Proceedings of Eurospeech ‘93, Berlin, September, pp.1657-1660, 1993.
18. Sperber, D. and Wilson, D. "Relevance, communication and cognition", Oxford, Basil Blackwell, 1986.