DESIGN, FORMALISATION AND EVALUATION OF SPOKEN LANGUAGE DIALOGUE

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

ABSTRACT

Dialogue model development is a major part of spoken language dialogue systems development. The dialogue model development process is a series of iterative interactions between design, formalisation and evaluation. This paper reports on the corpus-based development process of the dialogue model for the Danish dialogue system. The paper first describes dialogue model design through use of the Wizard of Oz method. Secondly, the continued formalisation of the dialogue model during the implementation phase is reported. The paper goes on to describe first results of the user test of the system, comparing these with the final results of the Wizard of Oz phase. Some issues for future work are raised in the conclusion.

1 INTRODUCTION

Dialogue model development is a major part of the development of spoken language dialogue systems (SLDSs). The entire development process, from the design of a first dialogue model through to the final user tests of the implemented system, may be viewed as a series of iterations, each iteration encompassing interacting aspects of design, formalisation and evaluation.

This paper describes how we addressed these interacting aspects when developing the dialogue model for the Danish prototype SLDS for domestic flight reservation.

The prototype SLDS, often simply termed the Danish dialogue system, has been developed in the Danish dialogue project which involves an effort of 30 man/years by the Center for Person-Kommunikation, Aalborg University, the Centre for Cognitive Science, Roskilde University, and the Centre for Language Technology, Copenhagen [Baekgaard et al. 1995].

The system runs on a PC and is accessed over the telephone. It understands continuous spoken Danish with a vocabulary of about 500 words and uses system-directed dialogue. The prototype runs in close-to-real-time. It consists of the main components shown in Figure 1. When a user calls the system, this will be detected by the *telephone line interface*. The *speech recogniser* then receives the user's speech signals. The speech recogniser is speaker-independent and uses HMMs to produce a 1-best string of words. The *parser* makes a syntactic analysis of the string and extracts the semantic contents which are represented in frame-like structures called semantic objects. The dialogue management module consists of the *ICM* and the *dialogue description*. The dialogue management module interprets the contents of the semantic objects and decides on the next system action which may be to send a query to the database, send output to the user, or wait for new input. In the latter case, predictions on the next user input are sent to the recogniser and the parser. The *database* contains information on timetables, flights, reservations and customers and rules for managing the information and queries it receives. System output is produced by concatenating pre-recorded phrases. The phrases are selected by the dialogue management module and replayed by a separate *reproductive speech module*. The *text recogniser* is only used when the speech recogniser is disabled, as has been desirable during debugging and test, cf. Sections 3 and 4. The *DDL-tool* is not part of the running system but is a tool used to create the dialogue description, i.e. the implemented dialogue model. The *Dialogue Communication Manager* is a data bus which transfers messages between all other modules.

The dialogue model for the system was iteratively designed by means of the Wizard of Oz method. The model resulting from the last WOZ iteration was implemented and debugged and the implemented system was tested with naive users.

The WOZ experiments produced a corpus of transcribed dialogues, user questionnaires, and inter-

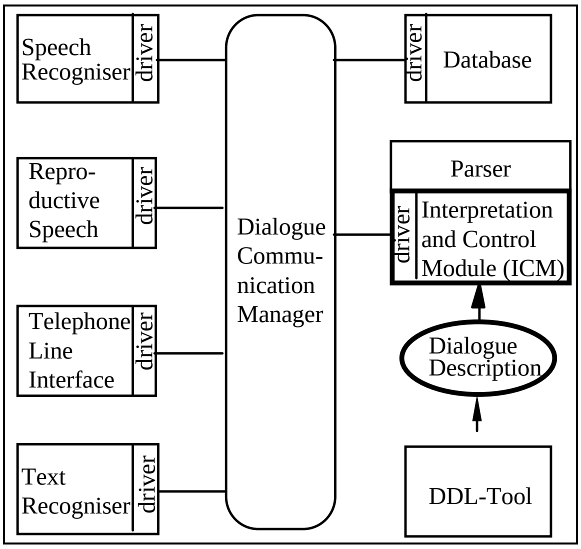

Figure 1. The overall architecture of the Danish dialogue system.

views; the implementation and debugging phase produced logfiles; and the user test produced logfiles and a corpus of transcribed dialogues, user questionnaires and interviews. Throughout the development process, these sources have served as a basis for evaluating the dialogue model by identifying user problems and revealing unsatisfied design goals and constraints.

The outcome of each evaluation cycle in the development process has served partly as a basis for improving the dialogue model and partly as input to the development of an applied theory of task-oriented dialogue. The evolving, formalised expression of the theory in its turn interacted with the dialogue design process. In addition, the dialogue design process as a whole has generated a consolidated series of guidelines for the design of usable SLDSs.

The remainder of this paper describes the dialogue development process for the Danish dialogue system in terms of iterative interaction between design, formalisation and evaluation based on corpora. Section 2 presents the WOZ experiments and the resulting corpus. Section 3 describes implementation and debugging. Section 4 reports on the user tests and their results. Section 5 summarises and concludes the paper.

2 DIALOGUE MODEL DEVELOPMENT

The Wizard of Oz (WOZ) experimental prototyping method is an iterative simulation technique which is well suited to the testing of dialogue models and the adjustment of design goals and design constraints prior to implementation. During each iteration a human (the ‘wizard’) simulates the system in dialogue with users who should preferably believe that they are speaking to a real system [Fraser and Gilbert 1991]. The dialogues are recorded, transcribed and analysed and results are used to improve the dialogue model. This iterative process continues until an acceptable dialogue model has been achieved.

2.1 The first dialogue model

The initial dialogue model was based on a number of different sources, including literature, field interviews with human travel agents and a standard timetable for Danish domestic flights which, in addition to departure and arrival times, contained information on i.a. fares and travel conditions. Two other important and intertwined sources were the technological constraints which were primarily imposed by the speech recogniser, and the goals to be achieved as regards usability [Dybkjær et al. 1993, Dybkjær et al. 1995a].

Since the application is based on access over the telephone, real-time performance was considered a constraint which had to be satisfied in order to obtain a usable system. However, this constraint, together with the chosen hardware, gave rise to new compulsory constraints caused by the speech recogniser:

- At most 100 words can be active in memory at a time for real time performance to be possible.
- 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 were also meant to maintain the recogniser error rate at an acceptable level.

Furthermore, because of limited project resources the system vocabulary size was set to about 500 words.

The main usability constraints, apart from real-time performance, were sufficient task domain coverage, robustness, natural forms of language and dialogue, and flexibility. These goals had to be traded off against the above resource and technological constraints. This was done during seven iterations of WOZ experiments.

2.2 The WOZ experiments

The first five WOZ iterations mainly served to train the wizard and adjust the dialogue model so that major shortcomings were repaired. Each WOZ iteration produced only a few dialogues. The dialogue model was initially represented as a loosely ordered set of predefined phrases. This made it difficult for the wizard to maintain consistency and quickly find an appropriate phrase. In addition, as the domain coverage was not yet complete, sometimes the needed phrase would not even be present in the dialogue model. To solve the wizard's problems we decided instead to use a graph structure for representing the dialogue model, cf. Figure 2. The graph has predefined system phrases in the nodes and expected contents of user input along the edges and turned out to significantly facilitate the wizard's job. Domain coverage was gradually made more complete. Users (subjects) were during this period exclusively system designers and colleagues.

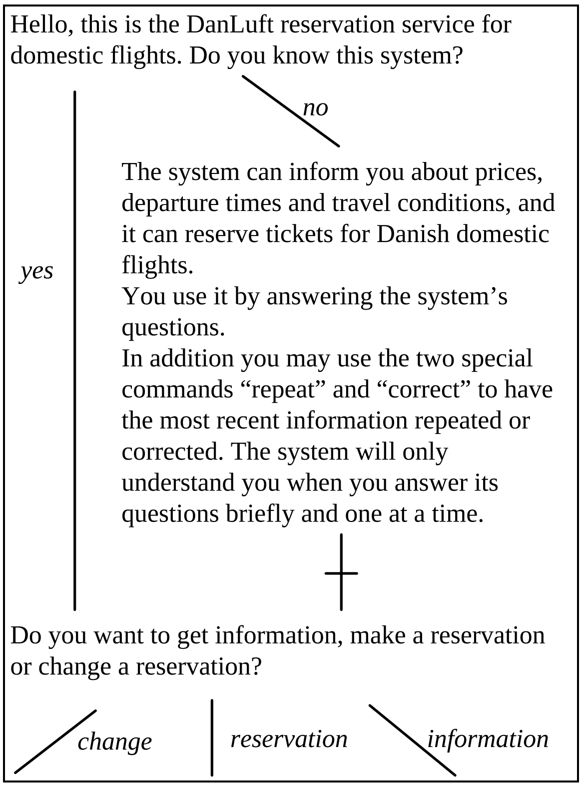

Figure 2. The introduction graph used in WOZ7 (translated from the Danish).

Throughout the experiments, interaction with the system was based on scenarios, i.e. domain-relevant tasks which the subject performed over the phone through dialogue with the system. 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 and training of the wizard. Most decisions on precise reservation details such as date of departure were left to the subjects. Subjects often revised a scenario or invented a new scenario on the fly which was never written down.

In the last three WOZ iterations a new set of scenarios was used. This second set included a total of 28 scenarios. Only some of them were used in WOZ5 whereas all were used in WOZ6 and WOZ7. 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. The scenarios from the second set contained more details than those in the first set and left few or no decisions to the subject. This facilitated the wizard's job because he would approximately know what a user would answer at a certain point during dialogue. However, the use of such detailed scenarios also had a negative effect in terms of users modelling the scenario phrases. This will be further discussed in Section 4 which also presents example scenarios.

The last two WOZ iterations were larger than the five first ones and were aimed directly at forming a basis for the dialogue model to be implemented and for the sub-language to be defined. Each of these two iterations involved 12 subjects. The majority of the subjects were external (non-in-house) and the rest were colleagues. Apart from three colleagues none of the subjects in the last two iterations 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.

Having agreed to participate, each subject in the sixth and seventh iterations received an envelope containing (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. Immediately before an experiment one of the system designers called the next subject at work and asked the subject to call the system. Subjects were not told in advance that the system was simulated. In a debriefing telephone interview after the session subjects were in WOZ7 asked whether they thought that they had interacted with a real system. The majority of external subjects believed that the system was real whereas the colleagues knew in advance that it was simulated.

The two last WOZ iterations each 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. 25 early dialogues were never transcribed. 24 different subjects had been used in the seven iterations.

For each iteration the recorded and transcribed dialogues were analysed and evaluated with focus on the extent to which the constraints and goals mentioned in Section 2.1 had been satisfied. Evaluation results were used as a basis for improving the dialogue model before the next WOZ iteration.

Between the fifth and sixth iteration we recorded a corpus of 25 Danish domestic flight reservation dialogues in a travel agency, corresponding to about one hour of spoken human-human dialogue. The original intention was to make these recordings early in the design process but due to practical problems this had not been possible. The structure of the WOZ6 dialogue model was adjusted in the light of typical task order structures identified in the human-human flight reservation dialogues.

2.3 The WOZ evaluation metrics

The evaluation metrics used during the WOZ experiments included measurement of the number of tokens (words) and types (different words), average utterance length, average number of utterances per dialogue exceeding 10 words, the longest turn, average number of turns per dialogue, number of user questions in per cent of the total number of turns (to converge towards zero), vocabulary size, cumulative word type/token ratio for subjects (to converge towards zero, only in WOZ7), average number of types per token in relation to number of tokens used by each subject (only in WOZ7), and the amount and nature of deviations from the normative model of how a scenario should be completed (only used systematically in WOZ6 and WOZ7, cf. below). The occurrence of user questions indicates that the user takes over the initiative. User questions therefore had to be eliminated as far as possible in order to satisfy the constraints on active vocabulary size and user utterance length. Convergence towards zero of the cumulative word type/token ratio is desirable because it indicates that the vocabulary size is sufficiently large for the application and that new users cannot be expected to use words out of the defined vocabulary.

As regards qualitative user evaluation of the system, subjects were asked to fill in a questionnaire, as mentioned above (from WOZ5 onwards). As indicated in parentheses above, some types of measurement were only made for the later WOZ iterations. In the early WOZ iterations some measurement results were much too far from the desired level and the material quite small, which made it irrelevant to study whether, e.g., the user type/token ratio converged.

In the last two WOZ iterations we compared the latest version of the system's dialogue model with the most recent, transcribed WOZ corpus in order to be able to systematically support improvements in system co-operativity. Each transcribed dialogue was plotted onto the graph structure which had system output in the nodes and expected contents of user utterances along the edges (cf. Figure 2). Deviations from the graph structure in terms of unexpected user or system behaviour were marked and the reason(s) for the behaviour analysed. When a deviation did not seem to have been caused by a wizard error, it was regarded as signifying a potential problem to be repaired.

Also at this stage, before each subsequent WOZ iteration we matched the scenarios to be used against the current dialogue structure in order to discover and remove potential user problems. This was done to some extent from WOZ4 onwards. The plotting and matching processes allowed identification of both actually occurring and potential user problems during dialogue. Actual user problems are such that actually occurred during user-system dialogue in the WOZ experiments. Potential user problems are problems discovered by the designers when putting themselves in the place of the users.

2.4 WOZ results

Each WOZ iteration produced quantitative as well as qualitative data. The quantitative data were used for measuring the extent to which the technological constraints were satisfied. Both quantitative and qualitative data were used for measuring usability constraint satisfaction. An important indicator of the degree of satisfaction of usability constraints is the number of user problems identified.

The technological constraints on maximum and average user utterance length were satisfied in WOZ7 (cf. Figure 3). Similarly, the task structure that had been developed appeared to make it possible to meet the constraint of a maximum active vocabulary of 100 words. This, however could only be achieved at the expense of user initiative. The dialogue model of WOZ7 was entirely system-directed, cf. Figure 4 [Dybkjær et al. 1993, Dybkjær et al. 1995a].

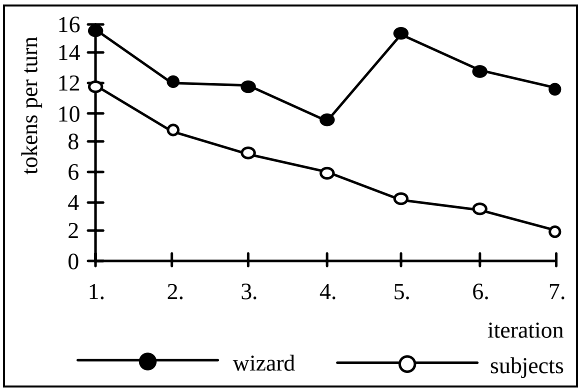

Figure 3. Average length of wizard and subject utterances in terms of tokens per turn.

The dialogue model was made system-directed by having the system conclude all its turns by a non-open question in order to preserve dialogue initiative. Non-open questions are questions which address a well-defined topic and ask for a specific piece of information. The non-open questions used by the system may be categorised as being of four types.

One type invites a yes/no answer, e.g.: “Do you want a return ticket?”

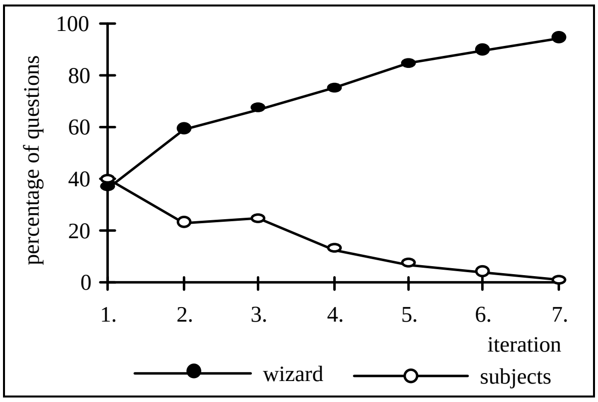

Figure 4. Number of questions in per cent of total number of turns.

The second type is a multiple choice question according to which the user is expected to choose an element from an explicit list of alternatives, for instance: “Is the ticket to be mailed or will the traveller pick it up at the airport?”

The third type of question invites the user to state a proper name or something similar, such as the name of an airport or an id-number. The application uses id-numbers instead of person names which cannot be dealt with because of vocabulary limitations. Users’ names are looked up in the database by using the id-number as key. For instance: “Please state the id-number of the traveller.”

The fourth type is the most open type or the one which allows the broadest variety of expressions in reply but which still concerns a specific topic, such as date of departure. For instance: “On which date will the journey start?”

None of these types of question invites the user to take over the initiative from the system.

During WOZ, a dialogue model was developed for ticket reservation as well as flight information and change of reservation. However, whereas "pure" reservation is a well-structured task, the information and change of reservation tasks are not. In a well-structured task there is a prescribed amount of information to be exchanged between the dialogue partners and the order in which this information is to be exchanged is often also prescribed to a certain extent. Complex ill-structured tasks such as the information task, on the other hand, are characterised by having a large number of optional sub-tasks. Each of these sub-tasks may be well-structured in itself but the overall task becomes ill-structured because of the optional character of the many sub-tasks it includes. This means that the system cannot make use of a valid stereotypical model that tells which sub-tasks the user wants to accomplish and possibly in which order [Bernsen et al. 1994a, Bernsen et al. 1994b, Dybkjær et al. 1995b].

Complex ill-structured tasks require mixed-initiative dialogue to be acceptable to users. Our heavy technological and feasibility constraints did not allow us to address the challenging task of designing mixed-initiative dialogue for a complex task such as the information task. It was therefore decided to implement only the reservation task which, because of its stereotypical structure allowed system-directedness and usability to co-exist. Thus, our recordings of human-human reservation dialogue in a travel agency showed that in reservation tasks the travel agent typically takes over after the initial customer turn and asks for the missing information piece by piece [Dybkjær and Dybkjær 1993].

As regards vocabulary size it was our hypothesis that 500 words would not be sufficient for the domain. The data from the WOZ experiments confirmed the hypothesis since the WOZ vocabularies did not clearly converge, not even the one in WOZ7. A 500 word vocabulary for the reservation task was defined mainly on the basis of the WOZ data. The user test of the implemented system was expected to provide more data on the sufficiency of the vocabulary.

With respect to evaluation of usability constraints, a large amount of work went into the identification and repair of actual and potential user problems. As mentioned in Section 2.3, we plotted transcribed dialogues onto the graph structure representation of the dialogue model and we matched scenarios against the dialogue model to be used next.

The work on identifying and repairing user problems was systematised at the end of the WOZ design phase. The user problems found during the entire WOZ experiment were analysed, classified and represented as violations, made by the dialogue system, of principles of co-operative dialogue. The result was a set of co-operative principles for human-machine dialogue derived from a WOZ corpus of realistic task-oriented (simulated) human-machine dialogue. Adherence to each principle should guarantee that a certain class of usability problems can be avoided in SLDS design more generally. [Bernsen 1993, Bernsen et al. 1994a, Bernsen et al. 1995b]

In order to have users evaluate the dialogue model, the WOZ subjects received a questionnaire, cf. Section 2.2. Figure 10 in Section 4.4 shows subjects' opinions of the dialogue system they had interacted with in WOZ7 and in the user test, respectively.

On the whole, subjects evaluated the system fairly positively in the WOZ questionnaires. The positive answers on robustness (few errors) and reliability in WOZ7 (see Figure 10) are probably due to the fact that the wizard did not simulate misrecognitions**.** In three cases there is no doubt that the WOZ7 system was evaluated negatively. Subjects found the system boring, perhaps because of the monotonous and slow voice used by the wizard in order to make subjects believe that they were interacting with a real system. Subjects also found the system inflexible and certainly the dialogue structure had become rigid and system-directed. Finally, it was quite clear that the subjects would prefer to talk to a human travel agent instead of the system. Probably the main reasons were the rigid dialogue structure and the correct impression that such a system has limited capabilities and cannot cope with non-routine matters.

Questionnaire results from the user test will be discussed in Section 4.4.

3 DIALOGUE IMPLEMENTATION AND DEBUGGING

3.1 Implementation

The reservation task was implemented in DDL (Dialogue Description Language) which is an event-driven recursive flow chart language [Dybkjær and Dybkjær 1994, Dybkjær et al. 1995a]. Compared to the initial formalisation of the dialogue task provided by the graph representation, the implementation task had to face two types of shortcoming. Firstly, some dialogue elements had not been simulated in the WOZ experiments at all and others had not been simulated in sufficient detail. Secondly, the graph representation was still far from possessing the formal rigour required of the implemented system and realised in the DDL flow chart representation. In more detail, the shortcomings were the following:

- *Task structure*. In the WOZ experiments only the structure of the hour task had been defined in some detail. The exact structure of other tasks had to be figured out during implementation. Moreover, in task-oriented dialogues most sub-tasks have a common basic structure and differ only on points such as the exact phrasing and the specific piece(s) of information they concern. This commonality had not been exploited in the WOZ graphs.
- *Meta-communication*. Focus in the WOZ graphs was on task communication, i.e. on turn-taking in the direct course of task execution. However, many turns in ordinary conversation are about the dialogue itself, i.e., they are turns of meta-communication. The possibilities of meta-communication were only rudimentarily expressed and never really used during the WOZ experiments.
- *Domain*. The different pieces of information, rules and constraints needed in the system's domain representation had no prior representation in the graphs, and the interface between domain representation and

dialogue was only implicit. For example, it had not been clearly defined which actions should be taken with respect to the system's domain representation when the user provided information on, e.g., the day of departure.

- *Dialogue state*. The overall as well as the local state of the dialogue had not been represented in the graph, including values of information slots, their status etc.

The above points had to be formalised during implementation through expanding and detailing the WOZ specification. The task structure required a new representation as described below, thus abandoning DDL's dialogues-as-graphs paradigm.

An outline of the main components of the implemented dialogue description or dialogue handler is presented in Figure 5 [Dybkjær et al. 1994].

The dialogue handling is task-oriented. There are two classes or levels of tasks:

- *Atomic tasks* concern one item of information, where an item is a value from the application or user domains. Atomic tasks are tagged with current system, user, and domain status, dialogue focus, and alternative values. Moreover, all user exchanges are done within atomic tasks, as explained below.
- *Compound tasks* manage the temporal structure of sets of atomic tasks. Examples are the reservation task and the overall dialogue frame. Compound tasks are modelled via a task dialogue structure represented as a graph where nodes are atomic tasks and edges are static links to other atomic tasks in a default template structure. The choice between links is made dynamically on the basis of Task Record and Dialogue History.

The atomic tasks follow a fixed scheme:

- check preconditions, i.e. if all required items are established;
- user-system exchange loop until item status is OK for both user and system:
  - ask the user:
    - for a value (and wait for answer), or
    - to select a value from a list (and wait for answer), or

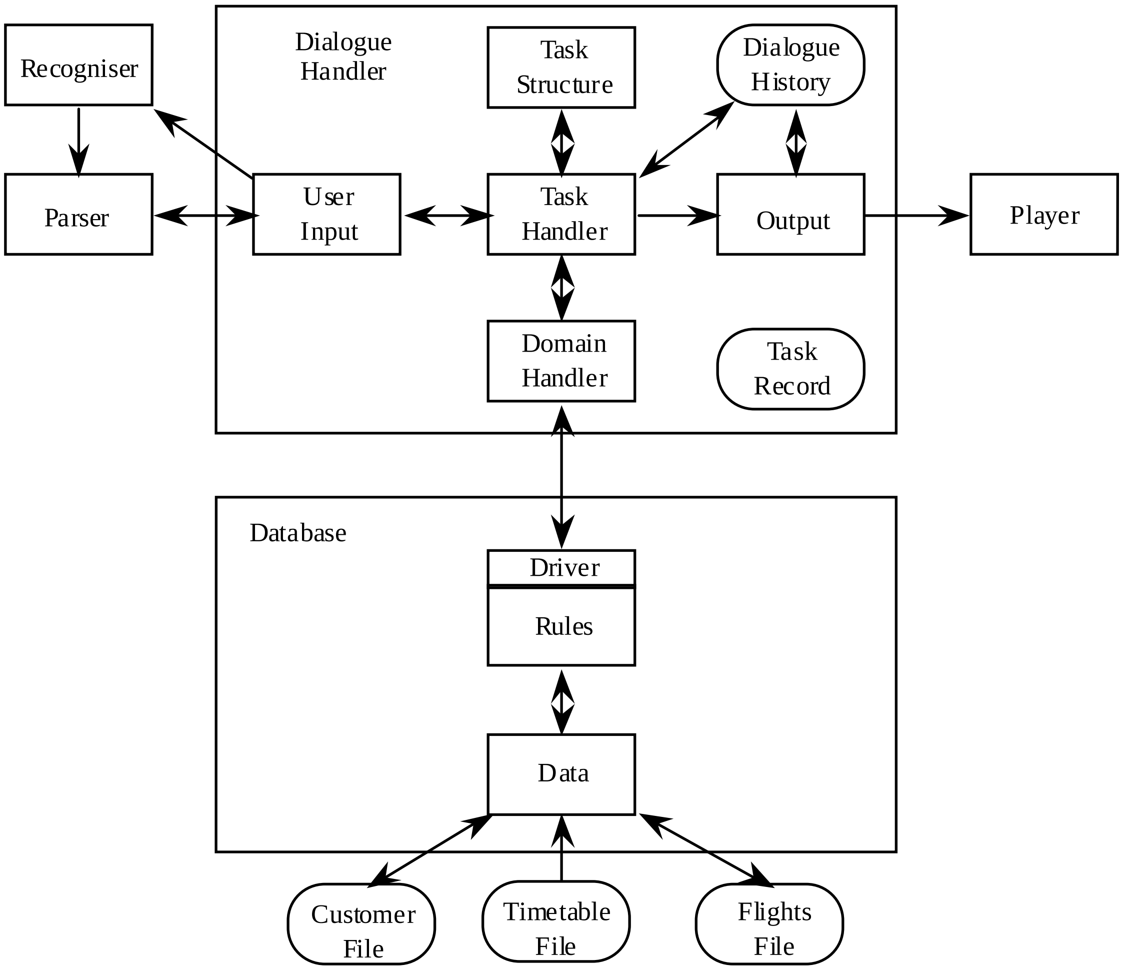

Figure 5. The communication structure of the recogniser, parser, player, dialogue handler and database with a detailed view of the dialogue handler and the database which represents domain knowledge. In the Dialogue Handler the Task Record is used by all processes. Rounded boxes indicate data and rectangular boxes indicate processes.

        - if a given value is desired (and wait for answer);
    - check the domain integrity of the value;
    - give feedback to users consisting in:
        - the accepted value, or
        - an error message;
- check post-conditions, i.e. if any other items are affected.

All checks and user-system exchanges are parametrised with respect to the items. In both pre- and post-checks and after user responses the Task Handler may jump directly to other tasks, thus circumventing the Task Structure.

As an example of the dialogue handling consider the following piece of dialogue in which the hour of departure is determined from S1b to S3a:

S0: On which date will the journey start?
U0: On Friday.
S1a: Friday May 19th.
S1b At which time of day?
U1: In the morning.
S2a: In the morning there are flights at 6:30 and 7:30.
S2b: Would you like one of these flights?
U2: Yes, 7:30.
S3a: 7:30.
S3b: On which date will the return journey start?

After S1a the Task Structure decides that the next item to be determined is the hour of departure. Control is transferred to the Task Handler which first checks if all other items required (route and date of departure) have been determined already. Then the exchange loop is entered and the system asks for time of day (S1b). The user answer (U1) is checked with the database which answers that there are two possible departures in the morning. In S2a this is given as feedback to the user. In S2b a new question is asked. The user answers the direct as well as the indirect question (U2). Since 7:30 has already been checked with the database, feedback is given without consulting the database again (S3a), the post-conditions are checked, control is transferred to the Task Structure, and a new cycle begins.

3.2 Debugging

A blackbox test was performed on the implemented dialogue model embedded in the entire system except the recogniser. The recogniser was disabled in order to make it possible to reconstruct errors. Internal communication between system modules was registered in logfiles. We created a number of test files all containing user input for one or more reservations of one-way tickets and return tickets with or without discount.

A test sequence always had to include an entire reservation involving several interdependent system and user turns. In a query-answering system a task will often only involve one user turn and one system turn. Hence one may ask a question and simply from the system answer determine if the system functions correctly for the test case. In a task such as ticket reservation which involves several turns, the system's reactions to the entire sequence of turns must be correct. An apparently correct system reaction, as judged from the system's immediate reaction, may turn out to have been partly wrong when we inspect the sequence of interdependent system reactions. Hence to test our dialogue model it was not sufficient to test, e.g., isolated transactions concerning customer numbers, possible destinations, or a selection of dates. Also the combinations of the test data had to be considered. Furthermore, each test reservation can only test a limited amount of cases so we had to create a long series of test reservations.

The blackbox test was not entirely exhaustive. In particular, it was not exhaustive as regards various interesting combinations of test data. However, the test did reveal a number of problems. Some of these were due to disagreements between the dialogue model specification and the implementation. But the majority of problems were such that had not been taken into account during specification.

Resources were not available for implementing solutions to all discovered problems. It was therefore considered, for each problem, how time consuming the implementation of a solution would be and how important it was. The hard problems were in many cases due to the fact that system-directed dialogue is not entirely sufficient to handle the cases in question. Solutions to such problems were not implemented because they would probably be sub-optimal anyway as long as the system-directed dialogue paradigm is maintained. Examples are round-trip tickets and reservations concerning, e.g., one passenger travelling out alone but going back together with another person. Both examples deviate from the standard reservation task and in the present system they have to be carried out as two separate reservation tasks. A round-trip ticket must be booked as two one-way tickets and the second example would have to be resolved by booking one return ticket and one one-way ticket.

The solutions which were implemented influenced not only the implementation but also the specification including the order of the dialogue structure. This again implied that the test files had to be revised to bring them in agreement with the specification. This is caused by the fact that the reservation task involves not only one user-system exchange but a whole sequence of exchanges which have to be made in a certain order.

The revised dialogue model was blackbox tested with the revised test files. Bugs were corrected but no major new unknown problems were revealed.

4 USER TESTS

When the system had been debugged we performed two series of user tests. In the first test the system was used with a simulated recogniser, in the second, the real recogniser was used. At the time of writing, the second test has not yet been completed and the analysis of results from the first test is in progress. Therefore, only first results from the simulated-recogniser user test are presented below. The setup and material used in the second test are the same as were used in the first test, cf. Section 4.1.

4.1 User test with a simulated recogniser

The system including a simulated recogniser was tested with naive users, i.e. users who had no previous knowledge of the system. A wizard keyed in the users' answers to a simulated recogniser. The simulated recogniser ensured that typos were automatically corrected and that input to the parser corresponded to an input string which could have been recognised by the real speech recogniser. The recognition accuracy would be 100% as long as users remained within the vocabulary and grammars known to the system. Otherwise, the simulated recogniser would turn input into a string which only contained words and grammatical constructions that were within the recogniser vocabulary and which conformed to the recogniser's grammar rules.

Ten external and two in-house subjects were used. Ten of them were secretaries. The percentage of secretaries approximately corresponds to the percentage of secretaries among the customers who called the travel agency in which we recorded our human-human dialogue corpus.

Each subject received an envelope containing (i) a letter informing on the experiment, (ii) a colour brochure introducing the system, (iii) four scenarios, and (iv) a questionnaire. The dialogues were conducted over the telephone as in the WOZ experiments. Immediately after interaction with the system, subjects received a telephone interview. In this interview all subjects stated that they believed that the system was real.

4.2 Scenario design

The two different sets of scenarios used in the WOZ experiments (Section 2.2) conform to the notion of *development* scenarios, i.e. scenarios which are intended to more or less systematically cover the intended system functionality and are normally designed by the system designers. Whereas the domain coverage of these scenarios was reasonable, meta-communication was not simulated. The scenarios did not give subjects incorrect information and subjects were not otherwise asked to simulate situations in which errors occurred. This proved to be a drawback during implementation since we had no information on users' meta-communicative reactions to work from. The conclusion is that the WOZ scenarios should have covered the same ground as should the input cases in a black-box test.

The scenario set used in the user test corresponds to the notion of *evaluation and test* scenarios. Based on the WOZ scenario experiences, we carefully considered what to test and why. We decided not to do user testing on a number of possible but unlikely cases of communication failure. These have been tested instead in the black-box test during system debugging. Since the flight ticket reservation task is a well-structured task in which a prescribed amount of information must be exchanged between user and system, it was possible to extract from the task structure a set of sub-task components, such as number of travellers, age of traveller, and discount vs. normal fare, any combination of which should be handled by the dialogue system. The scenarios were generated from systematically combining these components. This process generated a set of 20 scenarios.

The later WOZ experiments had shown that subjects tended to copy the temporal vocabulary used in the scenario descriptions, i.e. the expressions of date and hour of departure. Yet the sub-language vocabulary of the dialogue system was derived from the scenario-based WOZ dialogues. This constitutes a problem because a vocabulary defined on the basis of dialogues in which users model scenario phrases may not be sufficiently representative of realistic language use. On the other hand, scenarios clearly have to describe, to some necessary extent, the tasks to be performed by the subjects. It is not obvious, therefore, how one can avoid providing subjects with words or phrases which they will tend to repeat when answering the system’s questions, rather than selecting their own forms of expression

To explore how to avoid this effect and elicit a more realistic sublanguage, two groups of test subjects were formed each of which received a different version of the scenario material. One group received standard travel descriptions of the kind likely to be copied by subjects, whereas the second group received a new version of the scenarios in which the copying effect had been effectively blocked [Dybkjær et al. 1995c]. Each group consisted of six subjects.

We had carefully considered which information to mask in the scenarios, and how. For this purpose we used the categorisation of system questions into the four types mentioned in Section 2.4: yes/no questions, multiple choice questions, questions asking for a proper name or something similar, and questions asking for date or time.

The interesting point is that in the first three cases, the key information can only be co-operatively expressed in one of several closely related ways, which means that it does not matter if users model the expressions in the scenario representation. It is only in the fourth case that co-operative user answers may express the key information in many different ways. It is exactly in these cases that it is desirable to know how users would normally express themselves and hence important to prevent them from modelling the scenario representations. System questions in this case all concerned date and hour of departure. We therefore decided to concentrate on masking the scenario representations as regards date and hour of departure in order to avoid priming of the subjects.

In general, dates are either expressed in relative terms as being relative to, e.g., today, or in absolute terms as calendar dates. Hours are either expressed in quantitative terms, such as, e.g., ‘ten fifteen am.’ or ‘between ten and twelve’, or in qualitative terms, such as ‘in the morning’ or ‘before the rush hour’. The masked scenario representations never contained re-usable expressions referring to dates or hours of departure. Relative dates were expressed using a list of the days from today onwards. Absolute dates were expressed as calendar indices such as might be used by a customer when booking a flight. Quantitative hours were expressed using the face of a clock. Qualitative hours were expressed using (travel) *goal state* temporal expressions rather than departure state temporal expressions, for instance: ‘they want to arrive early in the evening’. This means that the subject, in order to determine when it would be desirable to depart, had to make an inference from the hour indicated in the scenario representation and generate a linguistic expression representing the result of that inference, thus excluding the possibility of priming.

All 20 scenarios were represented in two different versions. The masked version combines language and analogue graphics (cf. Figure 6) whereas the control group version uses standard linguistic text (cf. Figure 7) and roughly corresponds to the style of the second set of WOZ scenarios.

Jens and Marie Hansen (ID-numbers 1 and 4) and Steen and Jane Sørensen (ID-numbers 6 and 7) live in Copenhagen. They will attend a meeting in Aarhus as shown in the calendar which starts with today in boldface and shows the day of departure as the next day in boldface. The meeting starts and ends as shown on the two clocks. The flight takes about 35 minutes. The time to get from the airport to the meeting is about 45 minutes. The customer number is 4.

M T W T F S S M T W **T** F S S

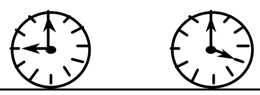

Figure 6. An analogue graphic scenario representation.

Jens and Marie Hansen (ID-numbers 1 and 4) and Steen and Jane Sørensen (ID-numbers 6 and 7) live in Copenhagen. They will attend a meeting in Aarhus on Thursday next week. The meeting starts at 9 am and ends at 4 pm. The flight takes about 35 minutes. The time to get from the airport to the meeting is about 45 minutes. Therefore they want the departure at 7:20 and, for the return journey, the departure at 17:30. The customer number is 4.

Figure 7. A text scenario representation corresponding to the graphic scenario of Figure 6.

4.3 User test evaluation metrics

The evaluation metrics used includes all relevant measurements, quantitative as well as qualitative, which were applied to the WOZ data. These measurements will allow comparison with the WOZ results and show if the user test data conform to the technological constraints which were the crucial parameters in the WOZ experiments. In addition, measurements have been made on meta-communication, system performance, other data pertinent to usability evaluation, and sub-language differences between the two groups of subjects which received different scenario versions.

4.4 Overall user test results

The user test produced a corpus of 57 dialogues. 32 dialogues were based on text scenarios and 25 dialogues were based on graphic scenarios. Subjects sometimes repeated a scenario if they did not succeed the first time. Figure 8 provides a translation from the Danish of a transcribed dialogue from the user test. The dialogue is based on the scenario shown in Figure 7.

The test results show that the average user utterance length is still within the prescribed limits, cf. Figure 9. The prescribed maximum utterance length was exceeded in 17 cases. 10 of these utterances were produced by the same subject. Particularly in the first dialogue this subject tended to repeat an utterance if the system did not answer immediately. However, the majority of long utterances, both for this subject and in general, was caused by user-initiated corrections which did not make use of the keyword ‘repeat’ but were expressed in free style by users. Two long utterances were produced by subjects who took over the initiative

_

S1: Hello, this is the DanLuft reservation service for domestic flights. Do you know how to use this system?

U1: No.

S2: The system can reserve tickets for Danish domestic flights. You use it by answering the system’s questions. In addition you may use the two special commands “repeat” and “correct” to have the most recent information repeated or corrected. The system will only understand you when you answer its questions briefly and one at a time. Please state your customer number.

U2: 4

S3: Customer number 4. How many people will travel?

U3: 4 people.

S4: 4 people. Please state the id-number of the first person.

U4: Id-number 1.

S5: Please state the id-number of the second person.

U5: Id-number 4.

S6: Please state the id-number of the third person.

U6: Id-number 6.

S7: Please state the id-number of the fourth person.

U7: Id-number 7.

S8: Id-numbers 1, 4, 6, and 7, Jens Hansen, Marie Hansen, Steen Sørensen, and Jane Sørensen. Where does the journey start?

U8: From Copenhagen.

S9: From Copenhagen. Where does the journey end?

U9: Aarhus.

S10: Aarhus. Do you want a return ticket?

U10: Yes please.

S11: The red discount is possible for certain departures on weekdays and the green discount is possible for weekend departures. If you choose red discount the return journey cannot start until two nights after the arrival date. Are you particularly interested in discount?

U11: No thanks

S12: On which date will the journey start?

U12: The 26th of January.

S13: Thursday the 26th of January. At which time of day?

U13: 7:20.

S14: 7:20. On which date will the return journey start?

U14: The 26th of January.

S15: Thursday the 26th of January. At which time of day?

U15: 17:30.

S16: 17:30. You have now booked 4 tickets for Jens Hansen, Marie Hansen, Steen Sørensen, and Jane Sørensen from Copenhagen to Aarhus on Thursday the 26th of January at 7:20 returning on Thursday the 26th of January at 17:30. Your reference number is 49. The price is 4760 kroner. Are the tickets to be mailed or will the travellers pick them up at the airport?

U16: We would like to pick them up at the airport, please.

S17: The tickets will be sent to the airport and can be picked up at check-in desk number 19. The tickets must be picked up 20 minutes before departure at the latest. Do you want anything else?

U17: No thanks.

S18: Goodbye.

–

Figure 8. A translated example of a dialogue from the user test.

when asked ‘Do you want anything else?’. Finally, in some cases subjects provided more information in their answers than had been asked for. All long utterances, therefore, were produced when subjects took over the initiative against the principles on which system-directed dialogue is based.

The dialogue is entirely system-directed as appears from the example in Figure 8, and this actually did prevent users from asking questions as was also the case in the later WOZ experiments. In the user test, only four out of 998 user utterances were questions. One question was asked because the subject had misread the scenario text. The three other user questions all concerned available departure times. This is not surprising since departure times constitute a type of information which users often do not have in advance but expect to be able to obtain from the system.

As predicted, the system’s vocabulary is not sufficient, in particular as regards quantitative time expressions, cf. Section 4.5.

The system’s task domain coverage is substantial but limitations exist exactly at points of maximum domain complexity where system-directed dialogue comes close to its limits, cf. Section 3.

Figure 10 compares answers from the WOZ7 questionnaires with answers to the user test questionnaires. In many cases there is no real difference between the two sets of answers. The negative development with respect to subjects' opinion on how easy it is to make corrections is probably due to the fact that misunderstandings were not simulated in WOZ7. This meant that hardly any meta-communication was required. In the user test, the simulated recogniser sometimes misunderstood what the user said. In addition, the use of keywords for making corrections does not form part of the natural human linguistic skills.

This concludes our presentation of the general data obtained in the user test. Additional data and a comprehensive analysis will be presented in [Bernsen et al. 1995a].

	WOZ7		User test	
Total number of subjects	12		12	
Total number of dialogues	47		57	
	User	System	User	System
Total number of turns	881	905	998	998
Total number of tokens	1633	10495	2468	12185
Total number of types	165	350	188	189
Longest turn	12	92	23	87
Total number of turns > 10 tokens	3	272	17	253
Average number of tokens per turn	1.85	11.59	2.47	12.20
Average number of types per turn	0.19	0.39	0.19	0.19
Average number of turns per dialogue	18.74	19.26	17.51	17.51
Average number of turns > 10 tokens per dialogue	0.06	5.79	0,30	4,44
Average number of tokens per dialogue	34.74	223.30	43.30	213.77
Average number of types per dialogue	3.51	7.45	3.30	3.32
Total number of questions	4	-	4	-
Number of questions in per cent of total number of turns	0.45	-	0.40	-
Average number of types per token	0.10	0.03	0.08	0.02

Figure 9. Comparison of results from WOZ7 and the user test. The number of system questions were not calculated. All system turns except for the closing phrase contained a question.

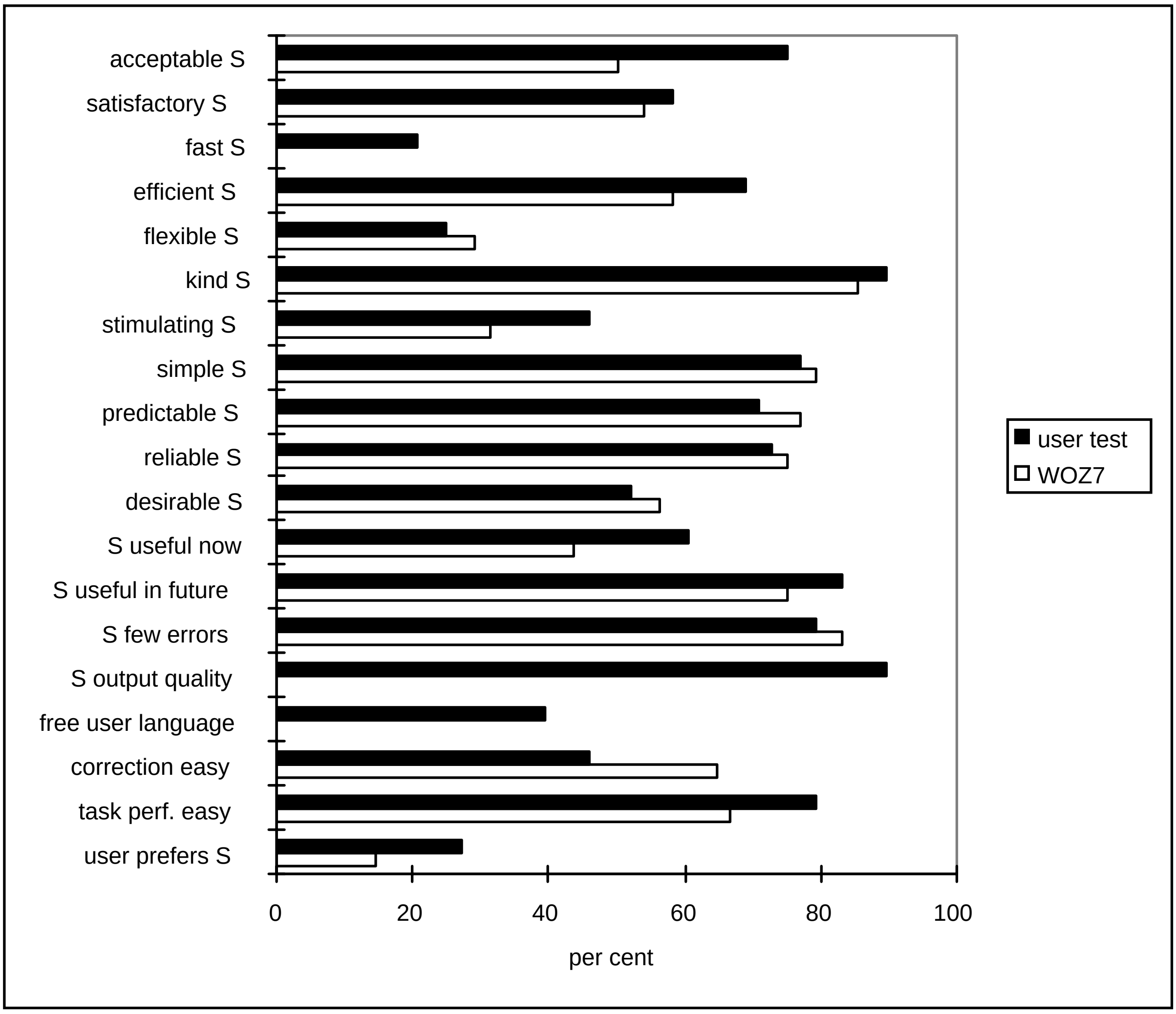

Figure 10. Subjects' answers to the questionnaires from WOZ7 and the user test in per cent of the maximum possible score. A score of less than 50 per cent indicates a negative opinion of the system. 'S' in the left-hand column refers to the system.

4.5 User test results related to scenario versions

The test results presented in Figures 9 and 10 above are based on analysis of the entire user test corpus. Figure 11 presents comparative data on the dialogues based on the graphic and text scenarios. Our hypotheses, as regards date and time, were that (1) there would be a massive priming effect from the text scenarios and none from the graphic scenarios, and (2) the dialogues based on graphic scenarios would contain a richer sub-language vocabulary than those based on text scenarios in terms of (i) total number of different words and (ii) out-of-vocabulary words. The first hypothesis was confirmed whereas the second was not. In addition, we had an unexpected result which could provide a strong argument in favour of using graphic scenarios for SLDS development.

4.5.1 Priming Effects

As expected, we found a massive priming effect from the text scenarios and virtually none from the graphic scenarios. The first row of Figure 12 expresses the "cleaned" number of user turns for which priming from the scenarios was possible. We have counted only the first occurrence of a user answer containing a date or a time in response to each of the four system questions concerning the dates and times of out and home journey departures. In these cases there is no immediate priming from the expressions used by the system itself and figures are not influenced by repeated or changed user answers.

Each date or time expression in the users' answers was compared to the scenario text. Complete matches and matches where *optional* parts of the date or time expression had been left out or added were counted at primed cases. If *non-optional* parts of the date or time expression had been changed, however, the case was considered as non-primed. For example, if the scenario said 'Friday the second of January' then 'the second of January' and 'Friday the second' would count as primed but not 'the second of first' which is a common Danish calendar expression.

	text scenarios	graphic scenarios
no. of subjects	6	6
no. of different scenarios	20	20
no. of dialogues	32	25
no. of user turns	547	451
no. of user turns*	181	178
no. of user tokens	1606	862
no. of user tokens*	705	451
no. of user word types	151	94
no. of user word types*	85	63
average user utterance length	2.94	1.91
average user utterance length*	3.90	2,53
longest turn	23	11
number of turns > 10 tokens	16	1

Figure 11. Data on the dialogues based on two different scenario types. * indicates that the figures only concern the dialogue parts on date and time.

	WOZ7	text	graphic
first date and time answers	74	106	84
primed answers	59	59	1
primed out date	91%	45%	-
primed home date	83%	23%	-
primed out hour	68%	78%	-
primed home hour	73%	71%	-

Figure 12. Priming effects in WOZ7, and for text and graphic scenario-based dialogues, respectively.

In the text scenario dialogues, priming was not equally distributed across date and time. This may have the following explanation. The time expressions used in the scenarios were similar to the feedback expressions used by the system and chosen from among the most common time expressions in Danish. A broader variety of date expressions was used in the text scenarios although most frequently of the form 'the second of January'. Furthermore, there are several frequent date expression formats. The system's feedback was of the form 'the second of first'. The decrease from 45% to 23% partly seems to be due to the fact that users changed from modelling the scenario text to modelling the system's feedback when answering the question about home date, and partly to the use of relative dates such as 'the same day'.

Throughout the WOZ scenarios the date format 'Friday the second of January' was used, which was in accordance with the system's feedback. This, and the general frequency of the expression, may explain the high date priming percentage in WOZ7.

4.5.2 Vocabulary Effects

The use of graphic scenarios did not result in a significantly richer vocabulary than use of the text scenarios, nor in the elicitation of more new words. On the contrary, dialogues based on graphic scenarios contained fewer different words, cf. Figure 11. The scenario sets generated no out-of-vocabulary dates and only nine new words for times.

Graphic scenario users massively replaced relative dates with absolute ones. This may be because people generally tend to do so on reservation tasks, or because people tend to do so in dialogue with machines which they know are inferior in language understanding. Whichever explanation is true, the effect is that subjects tended to standardise their date vocabulary by using exact dates rather than using their relative dates vocabulary.

Similarly, graphic scenario users tended to replace qualitative time with quantitative time, although less strongly so than when replacing relative dates by absolute dates. Again, the tendency is towards exactitude at the expense of using the language of qualitative time. The effect is another limitation on the vocabulary used.

We see three implications of these findings:

(i) The introduction, in SLDSs development, of graphic scenarios is not a means of doing away with good task scenario designs which may efficiently explore the task domain, users' language and user task performance. Good scenario design, however represented in the scenarios, is still essential to good dialogue design.

(ii) Given the fact that neither text nor graphic scenarios are able to elicit the full diversity of potential user language vis-á-vis the system, field trials of SLDSs developed by means of scenarios are still essential to the design of workable real-life systems.

(iii) The good news is that, in the graphic scenarios, subjects demonstrated a clear tendency towards expressing themselves in exact terms for dates and times.

4.5.3 An Unexpected Result

We found a significant difference in tokens (words) per turn between dialogues based on text and graphic scenarios, respectively, cf. Figure 11. Apart from the scenario representations, all subjects received identical material. They were asked the same questions, and they all believed that they communicated with a machine. Task contents were identical in the two sets of scenarios. There are no significant differences between the two user populations. The most plausible explanation, therefore, seems to be that the observed difference is produced by the different scenario representations themselves. In the text-based dialogues, subjects read aloud from their scenario representation. *They produce, in effect, spoken language which is not spontaneous, or which is not spoken discourse but read-aloud text.*

In the graphic-based dialogues, subjects cannot read aloud from their scenario representation because it does not contain textual expressions for date and time. To communicate the task contents of the graphic scenarios, subjects *have to* produce spontaneous spoken language.

When developing realistic SLDS applications, we need to copy or imitate realistic situations of use to the extent possible. Use of read-aloud text in communicating with the system is hardly close to realistic situations of use of most SLDSs. This would imply *that textual development scenarios which afford read-aloud solutions to communications with the system are unsuitable for SLDS development.* Other means of solution should be found in order to ensure that subjects do produce spontaneous spoken language in communicating with the system. One solution is to use analogue graphic representation of scenario sub-tasks when necessary. We have shown that this is possible and that it works for the representation of temporal scenario information.

5 CONCLUSION AND FUTURE WORK

Some preliminary conclusions on our dialogue model development process and the resulting dialogue system are:

The WOZ prototyping method is a powerful tool for dialogue model development although it does not eventually produce a model which is sufficiently formalised for implementation purposes. The quality of the produced model strongly depends on how well the simulations have been planned, trained, executed and iteratively evaluated. The main weaknesses of our own WOZ process were the lack of some form of tentative meta-communication simulation and the absence of formalisation details which therefore had to be developed during implementation. Overall, however, the WOZ development process has been successful in so far as there is reasonable correspondence between the final WOZ results and the results obtained during the user test.

The resulting dialogue system is entirely system-directed. This is primarily because of the strong constraints on active system vocabulary and user utterance length. A second important reason, however, is that we still lack a solid science base for developing mixed-initiative SLDSs for complex tasks [Bernsen et al. 1994b, Dybkjær et al. 1995b, Peckham 1993]. System-directedness makes task completion somewhat less efficient than might have been the case had mixed-initiative dialogue been feasible. As argued above, our corpora makes it clear that, for some sub-tasks of the reservation task, system-directed dialogue comes very close to its limits.

The system's qualitative time vocabulary is insufficient, as was expected. Its meta-communication apparatus, although functionally adequate, presents difficulties for novice users. However, users appear to quickly adapt to the system.

In addition to completing user testing and data analysis, we have begun to pursue two new directions of research. Both directions aim at consolidating a technologically and scientifically sound basis for building SLDSs for complex tasks. The first direction of research explores how the task of *informed reservation* might be formalised and implemented through the use of mixed-initiative dialogue [Dybkjær et al. 1995d]. An alternative to the use of mixed-initiative dialogue is to use multimodal technology. So, the second direction explores how the combined use of spoken input/output and graphic output may help overcome the limitations of system-directed dialogue in the performance of complex tasks.

ACKNOWLEDGEMENTS

The work described in this paper was carried out under a grant from the Danish Government's Informatics Research Programme whose support is gratefully acknowledged.

REFERENCES:

[Baekgaard et al. 1995] Baekgaard, A., Bernsen, N.O., Brøndsted, T., Dalsgaard, P., Dybkjær, H., Dybkjær, L., Kristiansen, J., Larsen, L.B., Lindberg, B., Maegaard, B., Music, B., Offersgaard, L., Povlsen, C.: The Danish Spoken Dialogue Project - A General Overview. *Proceedings of the ESCA workshop on Spoken Dialogue Systems*, Vigsø, Denmark, May 30 - June 2, 1995.

[Bernsen 1993] Bernsen, N.O.: Types of User Problems in Design. A Study of Knowledge Acquisition Using the Wizard of Oz. Esprit Basic Research project *AMODEUS II Working Paper RP2-UM-WP 14*, 1993. In Deliverable D2: Extending the User Modelling Techniques. June 1993.

[Bernsen et al. 1995a] 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 1995.

[Bernsen et al. 1995b] Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: Cooperativity in Human-Machine and Human-Human Spoken Dialogue. Submitted to *Discourse Processes*, 1995.

[Bernsen et al. 1994a] Bernsen, N.O., Dybkjær, L. and Dybkjær, H.: Task-Oriented Spoken Human-Computer Dialogue. *Report 6a from the Danish Project in Spoken Language Dialogue Systems*. Roskilde University, February 1994.

[Bernsen et al. 1994b] Bernsen, N.O., Dybkjær, L. and Dybkjær, H.: A Dedicated Task-Oriented Dialogue Theory in Support of Spoken Language Dialogue Systems Design. *Proceedings of the ICSLP Conference*, Yokohama, Japan, September 1994, 875-78.

[Dybkjær et al. 1993] 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 1993, 947-50.

[Dybkjær et al. 1995a] Dybkjær, H., Bernsen, N.O. and Dybkjær, L.: Dialogue Development and Implementation in the Danish Dialogue Project. To be published in *European Speech Projects*, Springer Verlag 1995 (in press).

[Dybkjær and Dybkjær 1994] Dybkjær, H. and Dybkjær, L.: Representation and Implementation of Spoken Dialogues. *Report 6b from the Danish Project in Spoken Language Dialogue Systems*. Roskilde University, May 1994.

[Dybkjær et al. 1994] Dybkjær, H., Dybkjær, L. and Bernsen, N.O.: Database Access via Spoken Language Interfaces. *Proceedings of FQAS '94*, Workshop on Flexible Query Answering Systems, Roskilde, November 1994, 69-79.

[Dybkjær et al. 1995b] Dybkjær, L., Bernsen, N.O. and Dybkjær, H.: Different Spoken Language Dialogues for Different Tasks. A Task-Oriented Dialogue Theory. To be published in *Human Comfort and Security*, Springer Research Report 1995 (in press).

[Dybkjær et al. 1995c] Dybkjær, L., Bernsen, N.O. and Dybkjær, H.: Scenario Design for Spoken Language Dialogue Systems Development. *Proceedings of the ESCA workshop on Spoken Dialogue Systems*, Vigsø, Denmark, May 30 - June 2, 1995.

[Dybkjær et al. 1995d] Dybkjær, L., Bernsen, N.O., Dybkjær, H. and Papazachariou, D.: On the Use of Context in Building Spoken Language Dialogue Systems for Large Tasks. *IJCAI Workshop on Context in Natural Language Processing*, Montreal, August 1995.

[Dybkjær and Dybkjær 1993] Dybkjær, L. and Dybkjær, H.: Wizard of Oz Experiments in the Development of a Dialogue Model for P1. *Report 3 from the Danish Project in Spoken Language Dialogue Systems*. Roskilde University, February 1993.

[Fraser and Gilbert 1991] Fraser, N.M. and Gilbert, G.N.: Simulating Speech Systems. *Computer Speech and Language* 5, 1991, 81-99.

[Peckham 1993] Peckham, J.: A New Generation of Spoken Dialogue Systems: Results and Lessons from the SUNDIAL Project. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, 1993, 33-40.