EVALUATION OF SPOKEN DIALOGUE SYSTEMS

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
phone: +45 46 75 77 11 fax: +45 46 75 45 02 url: http://www.cog.ruc.dk/

ABSTRACT

As spoken language dialogue systems (SLDSs) are taking off commercially, strong needs are being felt for improved methods and tools to support the evaluation of SLDS designs and products. Little is still known on dialogue evaluation and much work remains to be done. Based on development and evaluation of the dialogue component of an advanced SLDS, the paper reviews the evaluation procedures used and suggests improvements for use in future development projects. Concepts, methods and tools are described, results presented, and improvements proposed.

1. INTRODUCTION

The commercialisation of integrated spoken language dialogue systems (SLDSs) is a contemporary fact. Within the last few years SLDSs have matured to the point of attracting broad industrial interest and commercial SLDSs are now able to carry out routine tasks that were previously done by humans, thus generating significant savings in the companies or public institutions that install them. One of the most advanced systems currently in public use in Europe was introduced in 1994 by the Swedish Telecom Telia to automate part of the directory enquiries task [Forssten 1994].

Along with this development strong needs have arisen for effective evaluation procedures to be used during and after the development of SLDS products. In consequence, speech and natural language systems evaluation is emerging as a scientific subdiscipline in its own right. [Hirschman and Thompson 1996] Work on SLDSs evaluation has received significant stimulation from the ARPA Spoken Language Technology initiative [Galliers and Jones 1993, ARPA 1994] and progress is being made in Europe as well [Eagles 1995]. Progress and established methods exist for the objective evaluation of some of the individual components that make up SLDSs, such as speech recognition and speech synthesis, and objective evaluation procedures are beginning to appear for natural language parsing [Black 1996]. Still, evaluation of SLDSs today remains as much of an art and a craft as it is an exact science with established standards and procedures of good engineering practice. In particular, little is still known on dialogue evaluation including evaluation of dialogue components and integrated SLDSs. Thus,

- little is known about diagnostic evaluation [Hirschman and Thompson 1996], i.e. detection and diagnosis of errors, of dialogue components apart from traditional glass box and black box evaluation;
- little is known about systematic performance evaluation of dialogue components [Hirschman and Thompson 1996], i.e. measurements of the performance of the system in terms of a set of quantitative parameters;
- little is known about adequacy evaluation of integrated SLDSs [Hirschman and Thompson 1996], i.e., about how well a particular SLDS fits its purpose and meets actual user needs and expectations.

As SLDSs are being brought to the market, customer satisfaction becomes an important competitive parameter and hence an important element in measuring the success of an SLDS. However, user satisfaction does not necessarily derive from high technical performance, which only compounds the difficulty of SLDS adequacy evaluation:

> "From a commercial perspective, the success of a spoken dialogue system is only slightly related to technical matters. I make this somewhat bizarre pronouncement on the basis of first-hand practical experience. The key to commercial success is marketing: how a system is advertised to the end-users, how the system presents the company to those end-users, and how smoothly er-

rors are handled. I have, for example, seen trial systems with a disgracefully low word accuracy score receiving a user satisfaction rating of around 95%. I have also seen technically excellent systems being removed from service due to negative user attitudes." [Norman Fraser, personal communication.]

Other open research issues include:

- how to evaluate portability of systems across application domains;
- comparative performance and adequacy evaluation across SLDSs for different tasks. [Hirschman and Thompson 1996]

This paper presents a partial scheme for the evaluation of dialogue components and integrated SLDSs. It is based on the development and testing of the Danish dialogue system and includes suggested improvements, in terms of concepts, methods and tools, to the evaluation procedures that were actually applied during development and test of the system. Section 2 addresses evaluation of requirement specifications for SLDSs. Section 3 describes evaluation of dialogue model design. Section 4 describes evaluation of the integrated system. Section 5 concludes the paper. Evaluation of speech recognition and understanding components, and of language and speech generation components will not be discussed in what follows.

The Danish dialogue system is a ticket reservation system for Danish domestic flights. 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. The system is mixed-initiative, using system-directed domain communication and user-initiated, keyword-based meta-communication. The prototype runs in close-to-real-time. The system is a representative example of advanced state-of-the-art systems. Comparable SLDSs are found in [Aust and Oerder 1995, Cole et al. 1994, Eckert et al. 1995, Peckham 1993].

2. REQUIREMENT SPECIFICATION

The purpose of requirements specification is to list all the agreed requirements which the envisaged system should meet. There is no method which can ensure a complete and sufficient requirements specification. The craft and skills of experienced system developers are needed to make a qualified evaluation of imposed requirements. SLDS development and evaluation is still a relatively new field and there is no complete understanding of all the ingredients of SLDSs and their mechanisms of interaction. This adds to the difficulties of making a proper evaluation of an SLDS requirements specification. In the following we present experiences with establishing a requirements specification for the Danish dialogue system and proposals for its evaluation.

2.1 Realism criteria

The process of establishing a requirements specification for the Danish dialogue system was semi-realistic. The objective was to develop a realistic, application-oriented research prototype rather than a real application. This meant that we did not have real customers to talk to. However, we did have contact to a travel agency where we made interviews with travel agents and recordings of human-human reservation and information dialogues. The aim was to create a system which was realistic in the sense that it should meet, as far as possible, the needs and desiderata of potential customers. The system should offer economic advantage to potential customers and the choice of domain and technology should be reasonable in view of potential demands for SLDSs applications. For instance, it turned out to be a condition for launching the Danish dialogue project within the domain of telephone-based flight ticket reservation and information that a Danish parallel to the French Minitel did not exist at the time. Had such a system been in place, we had probably either chosen a different domain of application or a multimodal approach which included speech input/output. Another result of our considerations of application realism was that the system should be able to run on a PC so that Danish travel agencies could easily afford the needed hardware. Had we chosen more powerful equipment, the performance constraints on the system would have been less severe.

2.2 Feasibility and usability

The feasibility and usability constraints on the system to be developed may be illustrated as follows. Since the system should be accessed over the telephone, real-time performance was considered mandatory for the system to be usable. In the context of the chosen hardware, and given the limited capabilities that could be expected from the speech recogniser, the real-time requirement gave rise to additional constraints on active vocabulary size and user utterance length. Furthermore, because of limited project resources the system vocabulary size was set to about 500 words although this was likely to be insufficient given the chosen domain of application. This constraint, of course, would be meaningless in a commercial development context. In addition to real-time performance, the main usability constraints were: sufficient task domain

coverage, robustness, natural forms of language and dialogue, and dialogue flexibility.

2.3 Explicit requirements representation

As illustrated above, requirements behave as interacting constraints on the design process. This makes it desirable to create and maintain an explicit representation of the design space as it develops. If this is not being done, risks are that proper conclusions may fail to be drawn from interacting constraints with the result that the designers set out to what is in fact an internally conflicting task. We used the Design Space Development/Design Rationale (DSD/DR) approach to explicitly represent the evolving design space [Bernsen 1993b]. Several of the requirements mentioned above are represented in the DSD frame in Figure 1. A DSD frame represents the design space structure and designer commitments at a given point during system design. A series of DSD frames thus provides a series of snapshots of the developing design process. A DR frame represents the reasoning about a particular design problem (cf. Figure 5 in Section 4). It discusses the design options, constraint trade-offs and solutions considered and argues why a particular solution was chosen. Typically, there will be several DRs acting as links between two consecutive DSD frames. When combined with DR representations, DSD makes design space context and constraints explicit in support of reasoning, traceability and re-use.

We have had positive experience with using a DSD/DR representation in designing the Danish dialogue system. However, other methods of representation may be used instead. It is recommended to create an explicit requirements representation from the beginning of an SLDS development project. This is good engineering practice although often not followed with the result that is hard or even impossible to keep track of the design decisions that have been made and why they were made.

DSD No. N

A. General constraints and criteria

Overall design goal:

Spoken language dialogue system prototype operating via the telephone and capable of replacing a human operator;

General feasibility constraints:

Limited machine power available;

Scientific and technological feasibility constraints:

Limited capability of current speech and natural language processing;

Open research questions, e.g. research in dialogue theory;

Designer preferences:

Realism criteria:

The artifact should be preferable to current technological alternatives;

The system should run on machines which could be purchased by a travel agency;

The artifact should be tolerably inferior to the human it replaces, i.e., it should be acceptable by users while offering travel agencies financial advantage;

Functionality criteria:

Usability criteria:

Maximize the naturalness of user-interaction with the system;

Constraints on system naturalness resulting from trade-offs with system feasibility have to be made in a principled fashion based on knowledge of users in order to be practicable by users;

B. Application of constraints and criteria to the artifact within the design space:

Collaborative aspects:

Organisational aspects:

System aspects:

500 words vocabulary;

Max 100 words in active vocabulary;

Limited speaker-independent recognition of continuous speech;

Close-to-real-time response;

Sufficient task domain coverage;

Interface aspects:

Spoken telephone dialogue;

Task aspects:

User tasks:

Obtain information on and perform booking of flights between two specific cities;

Use single sentences (or max. 10 words);

Use short sentences (average 3-4 words);

System tasks:

User aspects:

User experience aspects:

C. Hypothetical issues:

Is a vocabulary of 500 words sufficient to capture the sublanguage vocabulary needed in the task domain?

D. Documentation:

E. Conventions:

DSD No. (n) indicates the number of the current DSD specification.

Figure 1. DSD representation which shows some major requirements for the Danish dialogue system. The actual DSDs constructed during the Wizard of Oz phase can be seen in [Bernsen 1993b].

2.4 Evaluation of specific SLDS requirements

If speech input and/or output are being considered for the application to be developed, evaluation is needed of whether speech is suited for the application given the evolving requirements specification. In case of a multimodal system one should also consider how well speech combines with other modalities considered for the system.

Whether speech is well-suited or not depends on properties such as the task, its structure and complexity and on whether the requirements derived from these properties are compatible with other requirements on, e.g., budget, time, reliability and technology. As mentioned above, the flight reservation and information tasks were found well-suited for a speech application. However, we had insufficient knowledge at the time for estimating the structure and complexity of the tasks as well as the resulting demands on the user-system dialogue. We thus began by designing mixed-initiative dialogue for reservation of flight tickets, change of reservation and information on departures, fares and travel conditions, and performed a series of Wizard of Oz (WOZ) experiments (Section 3). However, it turned out during the WOZ experiments that mixed-initiative dialogue was not feasible given the hard constraint on active vocabulary size, cf. Figure 1. Furthermore, change of reservation and, in particular, the information task which consisted of many different sub-tasks that could be combined in arbitrary order, were not well-suited for system-directed dialogue. For these reasons, the information task was never implemented. The change of reservation task which might have been feasible, with some difficulty, in system-directed dialogue, was not implemented because of resource limitations. With more knowledge early in the design process about task types and the dialogue types required by different task types, the information task might have been excluded much earlier. This task would have been evaluated as being non-feasible due to the conflict between the minimum requirements expressed in Figure 2 and the requirements specification.

Figure 2 was developed on the basis of our dialogue model design. Note that the figure is incomplete in several respects: it excludes systems that do not have speech (input) understanding, such as voice response systems and 'speech typewriters'; it does not consider the speaker-dependent/speaker-independent distinction; and user-directed dialogue needs more treatment. We have primarily compared relatively complex system-directed and mixed-initiative dialogue based on the distinction between well-structured and ill-structured tasks. *Well-structured tasks* have a stereotypical structure that prescribes which information needs to be exchanged between the dialogue partners to complete the task and, possibly, roughly in which order this is done. Such tasks may be acceptably managed through system-directed dialogue. *Complex ill-structured tasks* contain a large number of optional sub-tasks and hence are ill-suited for system-directed dialogue. Knowing, e.g., that a user wants travel information, does not help the system know what to offer and in which order. In such cases, some amount of user-directed dialogue or mixed-initiative dialogue would appear necessary to allow an acceptable minimum of usability.

2.5 Test criteria

Together with the requirements specification, performance and adequacy evaluation criteria should be established for the system to be developed. Early performance test criteria for the Danish dialogue system were the average and maximum user utterance lengths and the vocabulary size. We later discovered that we also needed a measurement for user initiative, cf. the discussion above. As a rough measure the number of user questions was used, cf. Section 3. Transaction success rate is a prime candidate adequacy evaluation criterion (Sections 3 and 4). Another possible criterion is the nature and number of interaction problems in a controlled scenario-based benchmark test (see Section 3). Subjective evaluation vehicles, such as questionnaires and interviews, are needed in addition to objective measures but it is very difficult to specify in advance the "scoring levels" that should be attained in questionnaires and interviews.

3. DIALOGUE DEVELOPMENT

Today's dialogue model design for SLDSs development is largely based on empirical techniques, such as the WOZ experimental prototyping method in which a person simulates (part of) the system to be designed [Fraser and Gilbert 1991] and, for simple dialogues, implement-test-and-revise procedures based on emerging development platforms. These techniques mainly build on designers' common sense, experience and intuition, and on trial and error. Whether WOZ is preferable to implement-test-and-revise depends i.a. on dialogue complexity and task domain and on risk and cost of implementation failure. WOZ is a costly method.

However, by producing data material on the interaction between a (fully or partially) simulated system and its users it provides the basis for early tests of the system and hence also for testing the coverage and adequacy of requirements. A number of different tests may be carried out on the material produced by WOZ experi-

Task complexity ->

Task type		**Task type**		**Task type**	
Small and simple tasks		Larger well-structured tasks Limited domains		Larger ill-structured tasks Limited domains	
Dialogue type		**Dialogue type**		**Dialogue type**	
Single-word dialogue		System-directed dialogue		Mixed-initiative dialogue	
Dialogue elements needed	**Other technology needed**	**Dialogue elements needed**	**Other technology needed**	**Dialogue elements needed**	**Other technology needed**
Either system or user initiative Limited system feedback	Isolated word recognition Small vocabulary No syntactic and semantic analysis Look-up table of command words No handling of discourse phenomena Representation of domain facts, i.e. a database Pre-recorded speech	System initiative in domain communication System feedback Static predictions System focus Dialogue act history Task record Simple user model Keyword-based meta-communication	Continuous speech recognition Medium-sized vocabulary Syntactic and semantic analysis Very limited handling of discourse phenomena Representation of domain facts and rules, i.e. expert knowledge within the domain Pre-recorded speech	Mixed user and system initiative System feedback Dynamic predictions System focus corresponds to user focus Linguistic dialogue history Dialogue act history Task record Performance record Advanced user model Mixed-initiative meta-communication	Continuous speech recognition Medium-to-large vocabulary Context dependent syntactic and semantic analysis Handling of discourse phenomena Representation of domain facts and rules, i.e. expert knowledge within the domain Representation of world knowledge to support semantic interpretation and plan recognition Speech synthesis

Figure 2. Increased task complexity requires more sophisticated dialogue to maintain an acceptable level of habitability. This again requires more and better technologies and increases the demands on dialogue theory and on the elements supporting the dialogue model. The figure shows minimum requirements.

ments. There is currently no agreement on which tests to carry out. We distinguish between three types of evaluation as mentioned in Section 1: diagnostic evaluation, performance evaluation and adequacy evaluation [Hirschman and Thompson 1996]. We shall also distinguish between objective evaluation and subjective evaluation. Diagnostic evaluation and performance evaluation are based on objective evaluation whereas adquacy evaluation include both objective and subjective evaluation.

The dialogue model for the Danish dialogue system was iteratively developed by the WOZ method. Seven WOZ iterations involving a total of 24 users were performed to produce the dialogue model which was implemented [Dybkjær et al. 1993]. The WOZ experiments produced a transcribed corpus of 125 scenario-based, task-oriented human-machine dialogues corresponding to approximately seven hours of spoken dialogue. We also collected a corpus of 25 human-human reservation dialogues in a travel agency. However,

we only used these dialogues to obtain information on the order in which the needed reservation details were achieved by the travel agent. At this level human-human dialogue parallels may serve as input to systems design. But the dialogues as such are much different from corresponding human-machine dialogues. Human-machine dialogues have to be much simpler than human-human dialogues because otherwise the system cannot handle them. Moreover, it is well-known that people tend to address computers in a way which is different from how they address humans, perhaps because of the systems' limited capabilities. For these reasons only human-machine data, such as those obtained through WOZ, are really reliable as a basis for a dialogue model.

3.1 Diagnostic evaluation

A major concern during WOZ is to detect and diagnose problems of user-system interaction. Eventually, we used two approaches, both based on the dialogue model representation, to systematically discover such problems. The dialogue model used in the WOZ experiments was represented as a complex state transition network that had system output in the nodes and expected contents of user utterances along the edges, cf. Figure 3.

The matching approach

One approach was to match, prior to each WOZ iteration, the scenarios to be used against the current dialogue model representation in order to discover and remove potential dialogue design problems. If a deviation from the state transition network occurred during the matching process, this would indicate a potential dialogue design problem which should be removed, if possible. Significantly, many problems were discovered analytically through these scenario-based designer walkthroughs of the dialogue model. This seems to be typical of dialogue model development and illustrates the need for a tool, such as a set of design guidelines, which could help designers prevent such problems from occurring.

The plotting approach

The second approach was to plot the transcribed dialogues onto the current dialogue model representation in order to systematically detect dialogue design problems from the interaction problems that occurred. As in the first approach, state transition network deviations indicated potential dialogue design problems. Deviations were marked and their causes analysed whereupon the dialogue model was revised, if necessary. Figure 3 shows an annotated sub-graph from WOZ6. The annotation shows that the user expected the system to confirm the commitments made. When it became clear that the system was not going to provide 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 stated the change, the dialogue continues (S is the simulated system, U is the user):

S7: Do you want to make other changes to this reservation?
U7: No, I don't.
S8: Do you want anything else?
U8: Ah no ...I mean is it okay then?
S9: [Produces an improvised confirmation of the change made.]
U9: Yes, that's fine.
S10: Do you want anything else?

From this point the dialogue finishes as expected. Analysis convinced us that the dialogue model had to be revised in order to prevent the occurrence of the user-initiated clarification meta-communication observed in U8, 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.

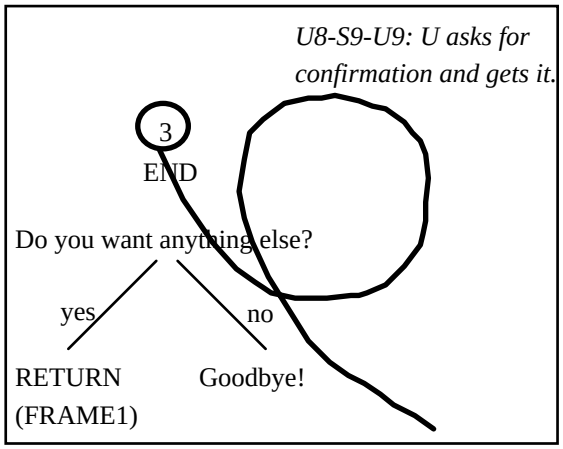

Figure 3. A plotted END sub-graph from WOZ6. The boldfaced loop that deviates from the graph path shows unexpected user dialogue behaviour which may reveal a dialogue design problem. The encircled number (3) refers to the point in the CHANGE sub-graph from which the experimenter jumped to the END sub-graph.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

Design guidelines

Many design errors were detected through use of the two above approaches. However, it would have been

preferable if we could have prevented these errors from occurring in the first place. Towards the end of WOZ we started to develop a tool which could serve the purpose of preventing interaction problems and which could be used no matter if WOZ is used or not.

All problems of interaction uncovered during WOZ were analysed and represented as violations of principles of cooperative spoken human-machine dialogue. Each problem was considered a case in which the system, in addressing the user, had violated a principle of cooperative dialogue. The principles were made explicit, based on the problems analysis. The WOZ corpus analysis led to the identification of 14 principles of cooperative spoken human-machine dialogue based on analysis of 120 examples of user-system interaction problems [Bernsen 1993a]. Each of the 14 principles was accompanied by a justification which served the additional purpose of clarifying its meaning and scope. If the principles were observed in the design of the system's dialogue behaviour, we assumed, this would serve to reduce the occurrence of user dialogue behaviour that the system had not been designed to handle.

The 14 principles of cooperative spoken human-machine dialogue were refined and achieved their present formulation as shown in Figure 4 through comparison with Grice's Cooperative Principle and maxims for cooperative human-human dialogue [Bernsen et al. 1996a]. Only SP10 and SP11 (on meta-communication) and the last part of GP10 were added later as a result of using the principles in analysing the dialogue corpus from the user test of the implemented system, cf. Section 4. The distinction between *principle* and *aspect* (Figure 4) is useful because an aspect represents the property of dialogue addressed by a particular principle. A generic principle may subsume one or more specific principles which specialise the generic principle to certain classes of phenomena. Although subsumed by generic principles, we believe that specific principles are useful to SLDS dialogue design. The principles are used by manually evaluating if each system utterance in isolation as well as in context violates any of the generic or specific principles. If it does, it is a potential source for communication failure which should be removed.

So far we have not had the opportunity to use the principles as design guidelines in an SLDS development process. However, we have successfully used them for evaluation purposes during the user test, as will be discussed in more detail in Section 4.

3.2 Performance evaluation

Between each of the seven WOZ experiments the dialogue model was evaluated and, based on the results, modified in order to achieve improved performance. The performance tests measured average and maximum utterance lengths, vocabulary size and convergence, and user initiative was roughly measured in terms of number of user questions. We also compared the results to those of earlier WOZ iterations in order to measure progress. The utterance lengths were eventually reduced to meet the requirements. The vocabulary, however, although sufficiently small within each iteration did not show convergence. Convergence towards zero of the cumulative word type/token ratio would indicate that the vocabulary size is sufficiently large for the application and that new users cannot be expected to introduce new words. However, as expected, a 500 words vocabulary turned out to be insufficient.

The early WOZ iterations allowed free mixed-initiative dialogue. We gradually transferred dialogue initiative to the system by letting the system ask questions of the user, thereby reducing the average user utterance length and the active vocabulary size. Much effort went into achieving a dialogue structure which corresponded to the one that users would expect based on their experiences from human-human reservation dialogues. Again this served to prevent the occurrence of user initiative. The domain dialogue was eventually made completely system-directed which turned out to be necessary in order to meet the constraint on active vocabulary size (Figure 1). Had we had the knowledge expressed in Figure 2 at the start of the WOZ experiments, we would have known already then that mixed-initiative domain communication would not be feasible.

3.3 Adequacy evaluation

We did not perform any objective adequacy evaluation of the WOZ material. However, it may be recommended to at least carry out evaluation of the transaction success. Although only based on simulated human-machine dialogue, such an evaluation may still provide valuable information on dialogue acceptability. The system should be implemented only when minimum requirements on transaction success have been met. Transaction success could thus serve as a stop criterion for WOZ. Transaction success is discussed in more detail in Section 4.

Subjective evaluation parameters

As user satisfaction is not just achieved through technically excellent systems and cannot be sufficiently measured through objective evaluation, it is important to collect users' opinions on the system

being developed at the earliest possible. WOZ provides a good basis for collecting users' opinions prior to system implementation, for instance through questionnaires and interviews. Questionnaires and interviews can be useful in identifying weaknesses that have been overlooked or cannot easily be identified through objective measurement. The difficulty with questionnaires and interviews is which questions to ask and how, and how to interpret the answers. Questionnaires also tend to be rigid, in particular if multiple choice is being used. If, on the other hand, questions are too open the risk is that people do not tell us what we would like to know. Also, people often do not like

Dialogue Aspect	GP no.	SP no.	Generic or Specific Principle
Group 1: Informativeness	GP1		Make your contribution as informative as is required (for the current purposes of the exchange).
	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.
	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.
	GP7	SP3	Provide same formulation of the same question (or address) to users everywhere in the system's dialogue turns.
	GP8 GP9		Be brief (avoid unnecessary prolixity). 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 cooperatively in dialogue. Ensure the feasibility of what is required of them.
	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		Take partners' relevant background knowledge into account.
	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		Take into account legitimate partner expectations as to your own background knowledge.
	GP12	SP8	Provide sufficient task domain knowledge and inference.
Group 7: Repair and clarification	GP13		Initiate repair or clarification meta-communication in case of communication failure.
	GP13	SP9	Provide ability to initiate repair if system understanding has failed.
	GP13	SP10	Initiate clarification meta-communication in case of inconsistent user input.
	GP13	SP11	Initiate clarification meta-communication in case of ambiguous user input.

Figure 4. The generic and specific principles of cooperativity in dialogue. Each specific principle is subsumed by a generic principle. The left-hand column characterises the aspect of dialogue addressed by each principle.

to spend time on writing about what they liked and did not like about the system. This is much easier to communicate in an interview. In interviews, however, subjects are rarely asked precisely the same questions in precisely the same way. This makes it even more difficult to compare user answers. In addition, people tend to express what they like and what they dislike in rather different ways.

In the last two WOZ iterations, we asked subjects to fill in a questionnaire after their interaction with the simulated system. In this questionnaire, users were first asked about their background, including how familiar they were with the task, with voice-responce systems and with systems understanding speech. They were then asked a number of multiple choice questions on the dialogue system. For each question they were asked to tick off one in five boxes on a scale from negative to positive, for instance 'difficult' versus 'easy'. The questions were the following: how was it to solve the tasks; what do you think of the number of errors made by the system; how was it to make corrections; how do you find the system now; would you prefer to call a travel agent or the system if you had the choice; what do you think of dialogue systems like this in the future; how well-prepared were you to use the system; how do you find the present system: rigid or flexible, stimulating or boring, frustrating or satisfactory, efficient or inefficient, desirable or undesirable, reliable or unreliable, complicated or simple, impolite or friendly, predictable or unpredictable, acceptable or not acceptable (all with the possibility of five choices). Finally, users were asked to provide free-style comments on whether something ought to be changed in the way in which users should address the system, what they liked about the system and what they did not like. On the average, users found the system rigid and boring and would prefer to talk to a human travel agent. Otherwise they were positive. The negative evaluation on the three points mentioned was not surprising given the rigid system-directed dialogue. The really valuable knowledge from a systems design point of view, however, was rather obtained through the free-style answers. In these, users would sometimes be very specific about what annoyed them when they used the system, thus providing us with clues to improvements.

We also interviewed users on the phone immediately after their interaction with the system. However, this was only to ask if they believed the system was real and to debrief them on the experiment.

4. THE IMPLEMENTED SYSTEM

The implemented system was subjected to the same tests as was the simulated system. In addition we measured transaction success and, based on the developed design principles presented in Section 3, we made a detailed analysis and evaluation of dialogue design problems. Also a blackbox test was carried out whereas a glassbox test was left out to save resources.

4.1 Glassbox and blackbox

There is no general agreement on the definitions of glassbox and blackbox tests. By a *glassbox test* we shall understand a test in which the internal system representation can be inspected. The test should make use of test suites that will activate all loops and conditions of the program being tested. The relevant test suites are constructed by the system programmer(s) along with an indication of which program parts the test suites are supposed to activate. Via test print-outs in all loops and conditions it is possible to check which ones were actually activated.

In a *blackbox test* only input to and output from the program are available to the evaluator. How the program works internally is made invisible. Test suites are constructed on the basis of the requirements specification and along with an indication of expected output. Expected and actual output are compared when the test is performed and deviations must be explained. Either there is a bug in the program or the expected output was incorrect. Bugs must be corrected and the test run again. The test suites should include fully acceptable as well as borderline cases to test if the program reacts reasonably and does not break down in case of errors in input. Ideally, and in contrast to the glassbox test suites, the blackbox test suites should not be constructed by the system programmer who implemented the system since s/he may have difficulties in viewing the program as a black box.

The dialogue model resulting from the seven WOZ iterations was implemented, as was the rest of the system. The dialogue model was, as mentioned, not subjected to a glassbox test whereas a blackbox test was carried out. The implemented dialogue model was embedded in the entire system except for the recogniser which was disabled to allow reconstruction of errors. Internal communication between system modules was logged in logfiles. We created a number of test suites all containing user input for one or more reservations of one-way tickets and return tickets with or without discount.

A test suite 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 suites had to be considered. Furthermore, because each test reservation can only test a limited amount of cases we had to create a long series of test reservations.

The blackbox test was not entirely exhaustive. 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. Each of the discovered problems were represented in a DR-frame along with a discussion of possible solutions, cf. Figure 5.

Resources were not available for implementing solutions to all the problems discovered. It was therefore considered, for each problem, how time consuming the implementation of a solution would be and how important it was. 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 suites had to be revised to bring them in agreement with the specification. The revised dialogue model was blackbox tested with the revised test suites. Bugs were corrected but no major new unknown problems were revealed.

4.2 User test with a simulated recogniser

A controlled user test of the implemented system was carried out with a simulated speech recogniser [Bernsen et al. 1995]. A wizard keyed in the users' answers into the simulated recogniser. The simulation 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

Design Project: P2		
Prepares DSD No. 8	**DR No.** 6	**Date:** 24.5.94
Design problem: No price information		
Users cannot get the price of the tickets they have reserved.		
Commitments involved		
1	It should be possible for users to fully exploit the system's task domain knowledge when they need it.	
2	Avoid superfluous or redundant interactions with users (relative to their contextual needs).	
Justification		
Only some users are interested in getting information on the price. Professional users loose time on an extra dialogue turn if they are asked whether they want it. On the other hand, for users wanting the price information this may be very important.		
Options		
1	Provide full price breakdown information at the end of a reservation task.	
2	Ask users if they want to know the price of their reserved tickets.	
3	Always inform users about the total price of their reservation (but not its breakdown into the prices of individual tickets).	
Resolution: Option 3		
There is a clash between the two design commitments because of the existence of different needs in the user population. Option 3 was identified and selected as a compromise between the two relevant design commitments. Option 3 does not require extra turn taking but mentions the price briefly.		
Comments		
Since P1 already computes the price it will be easy also to output this information to the user. It would be a possibility to allow the user to obtain additional price information (a breakdown into the prices of individual tickets) via the help function (see DR 12).		
Time estimate for developing and implementing solution		
Less than 1 day.		
Links to other DRs		
12 (help).		
Documentation		
Insert into next DSD frame		
Option 3.		

Status
Do the implementation.

Figure 5. A DR-frame for one of the problems detected during the blackbox test of the implemented dialogue model.

recogniser. 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.

A user test is meant to test if the system functionality expected by the user is present. A user test may be carried out as a controlled test or as a field test. In a controlled user test the users need not be those who will actually use the final system. However, it is recommended to select the test subjects from the target group to ensure that they have a relevant background. The background may influence the way in which people interact with the system. The tasks to be carried out (scenarios) are not selected by the participants in the controlled user test. To ensure a reasonable coverage of the test and representativity of scenarios and to bring it as close to benchmarking as possible, the scenario selection should ideally be made by an independent panel according to certain guidelines on, i.a., who should select the scenarios, their coverage of system functionality, number of scenarios per user and number of users. The panel should include end-users as well as system developers. A *field distribution problem* attaches to all results of controlled user tests. The frequency of different tasks across the domain of application may be different in real life from that imposed in the controlled user test. This may affect the frequency of different interactions problems.

In a field test real end-users are used as testers. The system to be tested is inserted in the environment in which it is supposed to work and is used. This means that the tasks carried out will be real-life tasks but will not necessarily be a representative selection unless the duration of the field test is very long. For reasearch systems the option of a field test will not always be available due to the missing customer. However, a controlled test may be preferable anyway because it allows an evaluation close to benchmarking.

The controlled user test of the Danish dialogue system was based on 20 different scenarios which had been designed by the system designers to enable exploration of all aspects of the task structure. Since the flight ticket reservation task is a well-structured task, 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.

Twelve external subjects who had never tried the system and who represented the target group, mostly professional secretaries, participated in the user test. Subjects conducted the dialogues over the telephone in their normal work environments in order to make the situation as realistic as possible. Each subject received an introductory letter, a brochure briefly describing the system, four scenarios and a questionnaire. Totally subjects were given 50 particular tasks based on 48 scenarios. A *task* consists in ordering one or more tickets for one route. The number of recorded dialogues was 57 because subjects sometimes reiterated a failed dialogue and eventually succeeded with the task. A *dialogue* is one path through the dialogue structure. As in the WOZ experiments, all dialogues were recorded and transcribed. In addition, all transactions between the individual system modules were logged. The test material was subjected to diagnostic evaluation, performance evaluation, and adequacy evaluation, the latter involving objective as well as subjective evaluation.

4.3 Diagnostic evaluation

The analysis of the user test corpus aimed at detecting problems of dialogue interaction. We did not use a state transition network representation of the implemented dialogue model as we had done for the simulated one. Instead, we used the following approach which not only considers deviations from the dialogue structure but also details of formulation. Based on the dialogue structure, a template was built which contained the system's questions. For each scenario, normative system questions and user answers were filled into the template. The key contents of the actual dialogues were then plotted into the template [Dybkjær et al. 1996]. An example is shown in Figure 6. A translation from the Danish of the problematic part of the dialogue is provided in the following:

S: Monday January 16. At which time?
U: 7:20.
S: There is no departure at 7:20. The closest other red departure is at 10:50. Do you want this departure?
U: No thanks.

[The above four turns are repeated another three times with the only difference that the user adds 'change' after 'no thanks' in two cases.]
S: At which time?
U: Change.
S: The journey starts on Monday January 16. Is that correct?
U: Yes.
S: At which time
U: 7:20.
S: There is no departure at 7:20. The closest other red departure is at 10:50. Do you want this departure?
U: Change.
S: The journey starts on Monday January 16. Is that correct?
U: Change.
S: Are you still interested in discount?
U: No.
S: The journey starts on Monday January 16. Is that correct?
U: Yes.
S: Departure at 7:20. Is that correct?
U: Yes.

Comparison between normative and actual system and user utterances and analysis of deviations which sometimes required use of the logged transactions between the system modules, led to the identification of three main classes of interaction problems: (1) linguistic problems, (2) problems of dialogue interaction and (3) other problems, such as cases of system breakdown. Dialogue interaction problems split into (a) dialogue design problems and (b) user errors [Bernsen et al. 1996b]. Only dialogue design problems are discussed in the following.

Each identified dialogue design problem was categorised according to which cooperativity principle, cf. Figure 4, had been violated. The following GPs and SPs were found violated at least once: GPs 1, 3, 5, 6, 7, 10, 11, 12, 13 and SPs 2, 4, 5, 6, 8, 10, 11. For classification purposes the principles were modified to express the cooperativity problem they represented, e.g. GP1: "Make your contribution as informative as is required (for the current purposes of the exchange)" was turned into "The system provides less information than required". Each problem was described in terms of its symptom (**S**), a diagnosis (**D**) was made and a cure (**C**) proposed, cf. Figure 7.

Scenario: G-1-4-a User: 2 Date: January 13 1995			
System questions	**Normative user answers**	**Actual user answers**	**Problems**
System already known	no / yes / -	yes	
Customer number	3	3	
Number of travellers	1	1	
ID-numbers	2	2	
Departure airport	Aalborg	Aalborg	
Arrival airport	Copenhagen	Copenhagen	
Return journey	yes	yes	
Interested in discount	no / yes	yes	
Day of departure (out)	January 16	Monday (January 16)	
Hour of departure (out)	7:20	7:20 (no departure) 7:20 (no departure) no, change [does not want one from list; change not caught by system] 7:20 (no departure) no [does not want one from list] 7:20 (no departure) no [does not want one from list] change [hour of departure] yes [out-day is January 16] 7:20 (no departure) change [hour of departure] change [day of departure] no [does not want discount] yes [out-day is January 16] yes [hour of departure is 7:20]	GP1, SP10 GP1 GP1 GP1 SP5 GP1
Day of departure (home)	January 16	Same day (January 16)	
Hour of departure (home)	17:45	17:45	

Delivery	airport / send	airport	
More	no	no	

Figure 6. Key contents of the expected (normative) and actual user-system exchanges in the dialogue G14a. In the third column key contents of the system's replies are indicated in parentheses unless they can be derived from the explanatory comments in square brackets. GP means generic principle and SP means specific principle.

S: S: Are you particularly interested in discount?. U: Yes please. ... S: At which time? U: 7:20. S: There is no departure at 7:20. The closest other red departure is at 10:50.
D: The system provides insufficient information. It does not tell that there is a blue departure at 7:20.
C: The system should provide sufficient information, e.g. by telling that there is no red departure but that there is a blue departure at the chosen hour.

Figure 7. Violation of GP1 in dialogue G14a. The system response is incomplete. It withholds important information and is therefore misleading. S is system and U is user.

The user test also served as a test of our cooperative principles and confirmed their broad coverage with respect to cooperative spoken user-system dialogue. Almost all of the 119 individual dialogue design problems identified in the user test material could be ascribed to violations of the cooperative principles. Only three additions had to be made to the principles established during WOZ. Two specific principles of meta-communication were added, i.e. SP10 and SP11 in Figure 4. Since meta-communication had not been simulated during WOZ and the WOZ corpus therefore contained few examples of meta-communication, this came as no surprise.

More interestingly, we had to add a modification to GP10, namely that it *should be feasible* for users to do what they are asked to do. For instance, in its introduction the system asks users to use the keywords 'change' and 'repeat' for meta-communication purposes and to answer the system's questions briefly and one at a time. Despite the introduction, a significant number of violations of those instructions occurred in the user test. For instance, users attempted to make changes through full-sentence expressions rather than by saying 'change'. Almost all of these cases led to misunderstanding or non-understanding. These violations of clear system instructions were initially categorised as user errors. However, upon closer analysis they were re-categorised as dialogue design problems. Although the system has clearly stated that it has non-normal characteristics due to which users should modify their natural dialogue behaviour, this is not cognitively possible for many users.

4.4 Performance evaluation

For the performance evaluation we measured the same parameters as in the WOZ experiments, i.e. the average and maximum utterance lengths, vocabulary size, and user initiative. The average user utterance length was still well within the required limits. However, the prescribed maximum user 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. 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 'correct' but were expressed in free style by users. Two long utterances were produced by subjects who took over the initiative when asked 'Do you want anything else?'. This question was clearly too open.

As predicted, the system's vocabulary was insufficient. The test corpus showed 51 out-of-vocabulary word types.

Subjects sometimes took over the initiative by providing more information than had been asked for and in four cases they asked questions. One question was asked because the subject had misread the scenario text. The three remaining 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. When users lack information, the reservation task tends to become informed reservation and hence an ill-structured task.

4.5 Adequacy evaluation

Adequacy evaluation should include measurement of transaction success. There is still no standard definition of "transaction success" [Giachin 1996]. In the Danish dialogue system we defined successes as reservations carried out according to the scenario specification or according to the user's mistaken interpretation of the scenario. As failures were counted reservations in which the user failed to get what was asked for even if this was due to an error committed by the user. Based on this definition, the task transaction success for the user test was 86% in that seven tasks were counted as transaction failures. One of the failures was exclusively caused by a user

who did not listen to the system's feedback and a second transaction failure was caused by a combination of a system problem (SP11) and a user error. The five remaining transaction failures were caused by system problems, i.e. violations of the principles GP5, SP2, SP4, SP5 and SP11, cf. Figure 4.

Misinterpretation of scenarios such as not asking for discount or ordering a one-way ticket instead of a return ticket is not a problem in real life. Nevertheless the situation is not desirable in a controlled user test since users carry out another scenario than they were asked to do which may affect system evaluation. A scenario which is not carried out may result in that part of the dialogue model remains untested.

An open question is whether transaction failures exclusively caused by user errors should be counted as failures or not. One may ask to which extent it is reasonable to blame the system for a failure.

One could also consider to use the result of the diagnostic evaluation of number and types of interaction problems as part of the adequacy evaluation. However, the problem is how to specify quantitative criteria in advance. It is not obvious how many and which types of interaction problems could be accepted.

Transaction success and number of interaction problems are not sufficient for measuring adequacy. For example, one cannot draw conclusions on user satisfaction from the transaction success rate nor from the number of interaction problems encountered.

Subjective evaluation parameters

To learn more on user satisfaction a subjective evaluation is needed. Therefore, also in the user test subjects were asked to fill in a questionnaire and received a telephone interview after interaction with the system. The questionnaire was very similar to one given to WOZ subjects. Only three questions had been added: how was the systems' speech; what do think of the language you used; was the system fast or slow. Output quality was rated high whereas subjects did not find that they could use free natural language. They found the system slow. These results are not surprising in view of the requirement to use keywords in initiating meta-communication, the missing sub-vocabulary parts, and the fact that the test used a bionic wizard system.

Many of the multiple choice answers were very similar to those from the WOZ questionnaires. Positive improvements over WOZ7 could be seen on acceptability, efficiency, usefulness and ease of task performance. There were also improvements in the evaluation of stimulatingness and preference of the system over a human travel agent but both were still low. The main reasons probably were the rigid dialogue structure and, in particular for the latter, the (correct) impression that the system has limited capabilities and cannot cope with non-routine matters.

There were drops in the positive evaluation on two important parameters, namely flexibility and ease of making corrections. The low evaluation on flexibility is probably due to the rigid, system-directed dialogue structure and the restriction to keywords for meta-communication. The negative development with respect to ease of making corrections is probably due to the fact that misunderstandings were not simulated in WOZ7. This meant that hardly any user-initiated meta-communication was required. In addition, the use of keywords for making corrections does not form part of the natural human linguistic skills.

Again as in WOZ, some useful and specific comments were given in reply to the open questions in the questionnaire. Although many subjects tended to write only one or two brief comments, a few subjects had bothered to write detailed and very useful replies.

In the telephone interview immediately after their interaction with the system users were asked the following four questions: How was it to talk to the system; what is your immediate impression of the sytem (specific problems/advantages); do you think the system was real; would you be interested in trying the system with the real recogniser. Like the free-style comments in the questionnaire, the telephone interviews provided inportant information on users' opinions of the system. The opinions expressed in the interviews were in accordance with the multiple choice answers in the questionnaire but contributed explanations of why the users held their opinions.

We did not ask the users to assign priority to their critical comments on the system. However, even if we had done this and modified the system accordingly, there would be no guarantee that users would then be satisfied with the system. User satisfaction is a conglomerate of many parameters, objective as well as subjective ones, cf. Section 1, and users may not even be aware of all the parameters which are important to them.

5. CONCLUDING DISCUSSION

This paper has addressed issues of SLDS evaluation as regards requirement specification, dialogue model design and the implemented and integrated system. Methods and tools used or developed during evaluation of the dialogue model of the Danish dialogue

system were presented and discussed. The presentation was structured in terms of distinction between diagnostic evaluation, performance evaluation and adequacy evaluation. In particular adequacy evaluation is difficult because it is not exclusively based on objective evaluation. Some of the test subjects were not at all interested in speaking to a computer system. This attitude may or may not change as speech systems become more common. Most people would probably be willing to use a speech understanding system provided that it is sufficiently attractive. However, what is considered attractive may vary from person to person. To some users, for instance, a mediocre system may become highly attractive if they receive a price reduction on tickets booked via this system.

Research is obviously needed on methods and tools which can support the three types of evaluation discussed. More research is also needed on aspects of evaluation which have not been addressed above. These include comparative systems evaluation, SLDS customisability evaluation, SLDS maintainability evaluation, strengths and limitations of speech functionality for different tasks, users, environments etc., speech and multimodality, and ergonomic aspects of speech applications.

ACKNOWLEDGEMENTS

The Danish dialogue system was developed in collaboration between the Center for Person-Kommunikation at Aalborg University (speech recognition, grammar), the Centre for Language Technology, Copenhagen (grammar, parsing), and the Centre for Cognitive Science, Roskilde University (dialogue and application design and implementation, human-machine aspects). The project was supported by the Danish Research Councils for the Technical and the Natural Sciences. We gratefully acknowledge the support.

REFERENCES

[ARPA 1994] ARPA. *Proceedings of the Speech and Natural Language Workshop.* San Mateo, CA: Morgan Kaufmann, 1994.

[Aust and Oerder 1995] Aust, H. and Oerder, M.: Dialogue control in automatic inquiry systems. *Proceedings of the ESCA Workshop on Spoken Dialogue Systems,* Vigsø, 1995, 121-124.

[Bernsen 1993a] 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.

[Bernsen 1993b] Bernsen, N.O.: The Structure of the Design Space. In Byerley, P.F., Barnard, P.J. and May, J. (Eds.): *Computers, Communication and Usability: Design issues, research and methods for integrated services.* Amsterdam, North-Holland, 1993, 221-244.

[Bernsen et al. 1995] 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, 1995, 1457-60.

[Bernsen et al. 1996a] Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: Cooperativity in human-machine and human-human spoken dialogue. *Discourse Processes,* Vol. 21, No. 2, 1996, 213-236.

[Bernsen et al. 1996b] Bernsen, N.O., Dybkjær, L. and Dybkjær, H.: User errors in spoken human-machine dialogue. To appear in *Proceedings of ECAI '96 Workshop on Dialogue Processing in Spoken Language Systems,* Budapest, 1996.

[Black 1996] Black, E.: Evaluation of broad-coverage natural-language parsers. *Chapter 13.4 in Cole et al. 1996.*

[Cole et al. 1996] Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A. and Zue, V. (editorial board), Varile, G. and Zampolli, A. (managing editors): *Survey of the State of the Art in Human Language Technology.* Sponsors: National Science Foundation, Directorate XIII-E of the Commission of the European Communities, Center for Spoken Language Understanding, Oregon Graduate Institute, 1996. http://www.cse.ogi.edu/CSLU/HLTsurvey/.

[Cole et al. 1994] 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, 1994, 683-686.

[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, 1993, 947-50.

[Dybkjær et al. 1996] Dybkjær, L., Bernsen, N.O. and Dybkjær, H.: Evaluation of Spoken Dialogues. User Test with a Simulated Speech Recogniser. *Report 9b from the Danish Project in Spoken Language Dialogue Systems.* Roskilde University, February. 3 volumes of 18 pages, 265 pages, and 109 pages, respectively, 1996.

[Eagles 1995] Eagles. Report of the spoken language systems working group 5. Technical report, EAGLES, EAGLES Secretariat, Istituto di Linguistica Computazionale, Via della Faggiola 32, Pisa, Italy 56126, Fax: +39 50 589055, Email: ceditor@tnos.ilc.pi.cnr.it, 1995. In press.

[Eckert et al. 1995] 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, 1995, 193-196.

[Forssten 1994] Forssten, B.: Speech technology: a one-shot possibility. *Proceedings of Voice'94,* London, October, 1994.

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

[Galliers and Jones 1993] Galliers, J.R. and Jones, K.S.: Evaluating natural language processing systems. *Technical report 291, University of Cambridge*

Computer Laboratory, March, 1993. To appear in Springer Lecture Notes in Artificial Intelligence.

[Giachin 1996] Giachin, E.: Spoken language dialogue. *Chapter 6.4 in Cole et al. 1996*.

[Hirschman and Thompson 1996] Hirschman, L. and Thompson, H.S.: Overview of evaluation in speech and natural language processing. *Chapter 13.1 in Cole et al. 1996*.

[Peckham 1993] Peckham, J.: A new generation of spoken dialogue systems: Results and lessons from the SUNDIAL project. *Proceedings of Eurospeech '93*, Berlin, 1993, 33-40.