The DISC Concerted Action

Niels Ole Bernsen and Laila Dybkjær
The Maersk Mc-Kinney Moller Institute for Production Technology
Odense University, Campusvej 55, 5230 Odense M, Denmark
emails: nob@mip.ou.dk, laila@mip.ou.dk
phone: (+45) 65 57 35 44 fax: (+45) 66 15 76 97

Abstract

This paper presents the aims and assumptions of DISC, the Esprit Long-Term Research Concerted Action No. 24823 “Spoken Language Dialogue Systems and Components. Best practice in development and evaluation” which starts on 1 June 1997. DISC will investigate a broad selection of state-of-the-art spoken language dialogue systems to identify current development and evaluation practice and pinpoint its deficiencies; and it will develop, test and disseminate a first detailed reference model of best practice procedures and methods, and a toolbox of associated concepts and software tools.

1. The need for best practice in development and evaluation

Software engineering best practice forms a backbone for the training of students in computer science and engineering who will later develop computer systems in industry and research. No current scheme specialises software engineering best practice to the particular purposes of dialogue engineering, that is, to the development and evaluation of spoken language dialogue systems (SLDSs). DISC aims to develop a first detailed and integrated set of development and evaluation methods and procedures (guidelines, checklists, heuristics) for dialogue engineering best practice as well as a range of support concepts and software tools. The goals of dialogue engineering include optimisation of the user-friendliness of SLDSs which will ultimately determine their rank among emerging input/output technologies. The methodology produced by DISC will contribute towards establishing dialogue engineering as a sub-discipline of software engineering, able to draw upon the rich variety of existing software engineering tools, methods and resources. Part of the DISC work will consist in codifying SLDSs best practice based on a detailed review and assessment of existing practices in the context of an overall model of software engineering best practice. Another part will be to develop concepts and software tools in support of SLDS development and evaluation. The DISC idea arose in the Elsnet (European Language and Speech Network) Research Task Group whose mission is to identify novel research goals that serve the integration of language and speech.

At this time there are no accepted standards or even widely understood benchmarks for assuring potential customers or users of SLDSs of the quality of systems. Neither are there any reliable methods for comparing the quality of two SLDSs before selecting one for deployment in the field. In an increasingly competitive marketplace, the ability to state that some system has been developed following a carefully designed and validated dialogue engineering methodology, along with the ability to report evaluation results in a standardised framework, is likely to give products developed in this way a competitive advantage. That in turn may stimulate take-up of the methodology by other organisations.

SLDS technology is taking off on a broad scale. For the year 2000, current estimates are that the global annual market for speech recognition alone will be $8 billion. According to the Ovum report on Voice Processing published last year, the global voice processing market in 1996 was $2.1 billion, and was expected to grow to $2.9 billion in 1997 and $3.75 billion in 1998. Even if, on a conservative estimate, only 1% could be described as SLDSs, that is still a very large number. The bulk of this business is taking place in the US but the opportunity to take a large and increasing share remains open to Europe.

Current commercial SLDSs are able to carry out routine tasks that were previously done by humans, thus generating significant savings in the companies or institutions that install them. During the last few years, interactive speech technology has begun significant deployment in real world applications in large vertical markets such as banking, finance and market research (Blyth and Piper 1994) as well as in telecommunications. In 1989 Bell Northern Research began deploying 'Automated Alternate Billing Services' through local telephone companies in the USA, with Ameritech being the first. The system rang customers, told them they had a collect call, and asked whether they would accept the call. Using a very small vocabulary (yes/no and some synonyms), the system successfully completed about 95% of the calls that were candidates for automation (Bossemeyer and Schwab 1991). In 1992, AT&T introduced a service to automate the other end of the transaction, allowing customers to place collect calls, use a calling card, order a person-to-person call, or place bill-to-third-number calls. User trials were considered successful, not just from a technical standpoint, but also because customers were willing to use the service (Franco 1993). By the end of 1993, it was estimated that over 1 billion telephone calls each year were being automated by this service. A key difference between the two systems is that the latter introduced word-spotting and barge-in technologies. A small but growing number of spoken dialogue services using these technologies have now been trialled by PTOs, mostly in the USA. These have focused on areas such as voice dialling, and directory assistance call completion. NYNEX thus has a system called VOIS in their public telephone system since 1990. It uses ASR to identify the number (the system asks for the number) that the customer has dialled but that for some reason was not valid or working. The system gives a spoken message why the connection did not occur (Ortel 1995). A European example is a system introduced in 1994 by Telia to automate part of the directory inquiries task (Forssten 1994).

An upcoming domain for advanced SLDSs is that of train information. Perhaps the most advanced SLDS in commercial use has been developed by Philips and is used by Swiss Rail. The system is based on the Philips Automatic Train Timetable Information System of which a demonstrator has been publicly available since February 1994, in Germany, on tel. +49 241 604020 (Aust et al. 1995, Aust and Oerder 1995). Similar train information systems are underway in the Netherlands, France and Italy. More advanced and flexible, large vocabulary SLDSs and systems integrating speech into multimodal systems are on their way from research laboratories to industrial exploitation and will have commercial significance by the end of DISC.

Publicly funded research has provided the major driving force for the technology advances exemplified by these systems. In the US, this has been coordinated by DARPA (latterly ARPA) through its competitive evaluations in large vocabulary speech recognition (Resource Management task) and spoken language understanding (ATIS task) (DARPA 1992; ARPA 1994). Europeans have been amongst the highest placed entrants in recent evaluations. There has been a clearer focus on the special issues associated with spoken language dialogue in Europe than in the USA. Projects such as SUNDIAL (Peckham 1993, Peckham and Fraser 1994, Fraser and

Thornton 1995, Peckham and Fraser forthcoming), the Danish national project on Spoken Dialogue Systems (Dybkjær et al. 1995, Bernsen et al. forthcoming), MAIS, RAILTEL (Lamel et al. 1995) and VerbMobil (Wahlster 1993) have established a strong base of expertise in Europe in spoken language dialogue systems.

Despite unquestionable progress, particularly in those parts of the SLDSs components field which have been delivering commercial applications for more than a decade, the design, development and evaluation of usable SLDSs is today as much of an art and a craft as it is an exact discipline with established standards and procedures of good engineering practice. Standard software design, development and evaluation practices can of course take development some way forward in terms of domain and task analysis, development languages, platforms, architectures and modularity, off-the-shelf components and state-of-the-art in some of the component technologies, such as speech recognisers and synthesisers, testing conformance with specifications etc. However, the remaining unknowns and undersupported development steps are evident from the following brief list of examples that derive from considering the development cycle as a whole, including the human factors aspects:

Project requirements and realism: whether to include spoken language dialogue in an application, given its task, domain, environment, user population and business requirements. Which input speech mode is needed for the application (single word vs. connected word vs. continuous speech; speaker dependent vs. speaker independent speech)? Is word-spotting sufficient? Which output speech mode is needed (speech synthesis, pre-recorded speech)? Can a natural and well-circumscribed sub-language be identified for the application? How far is an integrated resource containing domain and semantic knowledge needed and feasible? Can a modular, extensible and reusable architecture be found that will ultimately warrant the development costs of the first application? What are the minimal requirements on computational resources for the application?

Speech recognition and synthesis: how and to what extent can the speech recognition and grammar components cope with spoken language specificities, such as hesitations, repetitions of words or syllables, ill-formed phrases, incomplete sentences etc.; reject non-authorised words or interpret them using the context of the sentence or dialogue; and dynamically adapt to the user’s personal way of speaking (linguistic behaviour, own stereotypes etc.)? How to handle prosody in concatenated pre-recorded speech or in speech synthesis, given the application? Should different voice qualities be used for different information?

Language understanding and generation: whether to use stand-alone grammar and lexicon(s) or “hard-code” them into the system’s procedures. Use morphology (declarative and principled, but slow processing) or full-form lexicon (fast)? What is the best robust parsing scheme for the application? How integrate syntax and semantics? How efficiently separate resources from the procedures which use them (modularity)? How add linguistic knowledge (grammar and vocabulary) to the system during or after development (extensibility)? How to build one shared grammar for analysis and generation (modularity)?

Dialogue: how efficiently develop the dialogue model taking into account such aspects as dialogue type, dialogue strategies and minimal dialogue functionality needed for the application (e.g. system-directed, user-directed, mixed-initiative, use of dialogue history, inclusion of a user model); efficient error-handling mechanisms and strategies that counterbalance a less than 100% recognition rate; handling of awkward input and meta-communication design; usability of system communication with its users in context; system feedback design; dynamic adaptation within the task model to the course of the dialogue?

Development and evaluation, systems integration: in addition to the above: use Wizard of Oz (WOZ) or implement-test-and-revise - what are the trade-offs? What is needed for efficient WOZ design? Which corpus techniques to use for rapid characterisation of the domain and identification of expression variants? Which tools to use for the capture and analysis of data on user-system interaction? How to evaluate the application and its components in terms of such properties as relationship between component performance and overall system performance; voice quality and system wordings in dialogue; assessment of the effects of speech recognition errors on spoken language understanding and dialogue flow; spoken sub-language adequacy (lexicon and grammar) for language understanding and generation; robustness of parsing and error recovery; transaction success; beyond crude measures of dialogue quality, such as duration, number of turns or error counts to the identification of interaction problems, their types, severity and remedies; questionnaire design; correlation of errors with human ratings; user satisfaction and speaker style; extensibility, modifiability, adaptability; getting beyond the ARPA-style comparative SLDS assessment methodology using response comparison, which is a relatively crude measure?

2. The DISC objectives

DISC will pursue the following specific objectives:

- To carry out a detailed review and investigation of existing practices for a wide range of SLDSs and components development and evaluation.
- To define a detailed current best practice scheme of methods and procedures for SLDSs and components development and evaluation.
- To develop to the stage of industrial applicability a range of concepts, methods and software tools based on ideas and preparatory work at the partner sites.
- To test methods, procedures, concepts and software tools on industrial and applied academic development projects to the extent feasible within the duration of the Action.

The DISC partners are: The Maersk Mc-Kinney Moller Institute for Production Technology (MIP), Odense University, Denmark (coordinator); Human-Machine Communication Department, Centre National de la Recherche Scientifique (CNRS-LIMSI), France; Institut für Maschinelle Sprachverarbeitung (IMS), Universität Stuttgart, Germany; Department of Speech, Music and Hearing, Kungliga Tekniska Högskolan (KTH), Sweden; Vocalis Ltd,UK; Daimler-Benz, Germany; Stichting Elsnet, The Netherlands.

All partners contribute to the Action access to products and running prototypes and their components as well as to prototypes under development. DISC will take advantage of existing practices, theories and tools, including results of the US ARPA exercise in comparative SLDS evaluation, as well as emerging results in the fields of de facto standards and guidelines for speech products, natural language components and evaluation from LRE EAGLES and experience from national initiatives in component evaluation methodologies, such as the German Morpholympics and the French GRACE project and other evaluation actions of the AUPELF group.

The envisioned industrial benefits of DISC will be:

- Progress towards the integration of SLDSs best practice into software engineering.
- Improved feasibility assurance of development projects (risk minimisation) and more exact feasibility assessment.
- Improved procedures, methods, concepts and software tools.

- Reduced development costs and time, improved maintenance and reusability.
- Improved product quality and increased flexibility and adaptability.
- Progress towards the establishment of dialogue engineering standards.
- Improved guarantees to end-users that a product has been developed following best software and cognitive engineering practice. Enabling end-users to objectively assess different systems and components technologies against one another and choose the right product according to quality, price and purpose.

DISC will achieve its stated goals through the five work packages shown in Figure 1. WP1, WP2 and WP3 cover the main results-building activities. These WPs will focus on a set of aspects of SLDSs including speech recognition, speech generation, language understanding and generation, dialogue management, human factors and systems integration.

To ensure common approaches to each of the main results-building activities, and to ensure cross-aspect compatible results, each approach will have to include a set of agreed *evaluation criteria.* Thus, (a) the common approach for mapping out current practice includes criteria for the evaluation of current practice; (b) the common approach for testing best practice methods and procedures on industrial cases includes criteria for evaluating the transferability success of these methods or procedures; and (c) the common approach for iterative development and testing of novel concepts and software tools includes criteria for deciding the feasibility of specific development and testing projects as well as for evaluating transferability success.

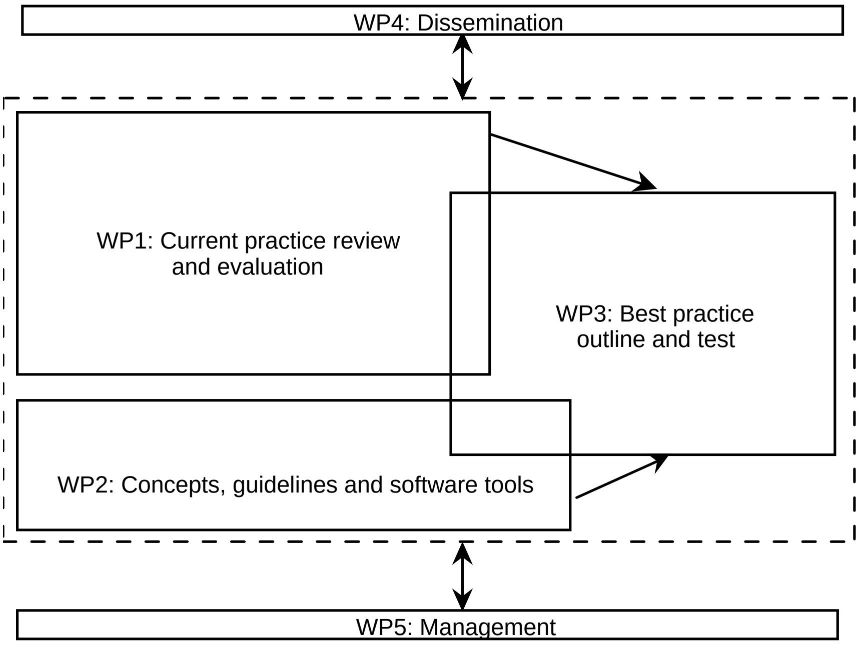
WP4: Dissemination
WP1: Current practice review and evaluation
WP3: Best practice outline and test
WP2: Concepts, guidelines and software tools
WP5: Management

Figure 1. The 5 DISC work packages (WPs) and their interrelations. The relative sizes of the boxes roughly correspond to the relative amount of effort allocated to each WP.

3. Dialogue design: an example of tool development in DISC

One of the development and evaluation tools to be completed in DISC is a set of guidelines for the design of cooperative human-machine dialogue. The tool was developed in the course of designing, implementing and testing the dialogue model for the Danish dialogue system. The first set of guidelines was based on problems of dialogue interaction observed in the Wizard of Oz corpus produced during systems specification and design. Each observed problem was considered a case in which the system, in addressing the user, had violated a guideline of cooperative dialogue. Corpus analysis led to the identification of 14 guidelines of cooperative spoken human-machine dialogue based on 120 examples of user-system interaction problems. If those guidelines were observed in the design of the system's dialogue behaviour, we assumed, this would increase the smoothness of user-system interaction, reduce user-initiated meta-communication for clarification and repair, and improve user satisfaction with the system.

The guidelines were refined and consolidated through comparison with an established body of maxims of cooperative human-human dialogue (Grice 1975) which turned out to form a subset of our guidelines. The resulting 22 guidelines were grouped under seven different *aspects* of dialogue, such as informativeness and partner asymmetry, and split into *generic* guidelines and *specific* guidelines. A generic guideline may subsume one or more specific guidelines which specialise the generic guideline to a certain class of phenomena. Figure 1 shows shortform versions of the guidelines.

Dialogue Aspect	**GG**	**SG**	**Generic or Specific Guideline**
Group 1: **Informativeness**	GG1		*Say enough.
		SG1	State user commitments explicitly.
		SG2	Provide immediate feedback.
	GG2		*Don't say too much.
Group 2: **Truth and evidence**	GG3		*Don't lie.
	GG4		*Check what you will say.
Group 3: **Relevance**	GG5		*Be relevant.
Group 4: **Manner**	GG6		*Avoid obscurity.
	GG7		*Avoid ambiguity.
		SG3	Ensure uniformity.
	GG8		*Be brief.
	GG9		*Be orderly.
Group 5: **Partner asymmetry**	GG10		Highlight asymmetries.
		SG4	State your capabilities.
		SG5	State how to interact.
Group 6: **Background knowledge**	GG11		Be aware of users' background knowledge.
		SG6	Be aware of user inferences.
		SG7	Adapt to novices and experts.
	GG12		Be aware of user expectations.
		SG8	Cover the domain.
Group 7: **Repair and clarification**	GG13		Enable meta-communication.
		SG9	Enable system repair.
		SG10	Enable inconsistency clarification.
		SG11	Enable ambiguity clarification.

Figure 1. Guidelines for cooperative system dialogue. GG means generic guideline. SG means specific guideline. The guidelines are expressed in shortform. Fullform expressions are found in (Bernsen et al. 1997). The generic guidelines are at the same level of generality as are the Gricean maxims (marked with an *). Each specific guideline is subsumed by a generic guideline.

The consolidated guidelines were then tested as a tool for the diagnostic evaluation of a corpus of 57 dialogues collected during a scenario-based, controlled user test of the implemented system. The availability of the user scenarios meant that problems of dialogue interaction could be objectively detected through comparison between the contents of expected and actual user-system exchanges. Each detected problem was (a) characterised with respect to its *symptom*, (b) a *diagnosis* was made, sometimes through inspection of the log of system module communication, and (c) one or several *cures* were proposed for repairing system dialogue behaviour. The diagnostic analysis may demonstrate that new guidelines of cooperative dialogue design must be added to the existing body of guidelines. We found that nearly all dialogue design errors in the user test could be classified as violations of our guidelines. Two *specific* guidelines on meta-communication, SG10 and SG11, had to be added, however. This was no surprise as meta-communication had not been simulated and therefore was mostly absent in the WOZ corpus.

To test and increase the generality of the tool, we have applied it as a dialogue design guide to part of a corpus from the Sundial project (Peckham 1993). The corpus comprises close to 100 early WOZ dialogues in which subjects seek time and route information on British Airways flights and sometimes on other flights as well. We selected 48 dialogues three of which were used for training. The remaining 45 dialogues were independently annotated and analysed by two experts in using the tool (A1 and A2) and one novice (A3). Each system utterance was analysed in isolation as well as in its dialogue context to identify violations of the guidelines.

Ideally, this test would increase the generality that can be claimed for the tool in four ways: (1) the *system dialogue* is different from that of the Danish dialogue system (mixed initiative vs. system directed); (2) the *task type* is different (information vs. reservation); (3) the tool is being used as an early *dialogue design guide* rather than for diagnostic evaluation*;* and (4) *circumstances* are different because we do not have the scenarios used in Sundial. If the tool works well under circumstances (4), we shall know how to use it for the analysis of corpora produced in, e.g., field tests with implemented systems in which scenarios are entirely absent.

Applying the tool to the Sundial corpus led to the identification of a large number of dialogue design problems all of which could be classified as violations of existing guidelines. Thus, the different system dialogue (1) and the different task type (2) compared to the Danish dialogue system did not reveal a need for additional guidelines.

Using the tool as an early dialogue design guide (3) is not significantly different from using it for diagnostic evaluation. The main difference is that early WOZ dialogues appear to produce more, and often more complex, guideline violations. In the 45 trial dialogues, the two experts found and agreed on 354 violations in the system's utterances. Many of these violations were complex in the sense that one system utterance violates several guidelines. In the user test corpus from the Danish dialogue system, experts A1 and A2 found and agreed on 117 violations in 57 dialogues, and at most two different guidelines were found violated in the same utterance. We find it likely that complex violations occur less frequently in corpora from later systems development phases. This hypothesis will be tested on other SLDS corpora during DISC.

The important generalisation (4) poses a particular problem. When, as in controlled user testing, the scenarios used are available, it is relatively straightforward to objectively detect dialogue design errors. However, when this is not the case, the problem arises of whether the corpus analysers are actually able to detect *the same* problems in a dialogue prior to classification. In the Sundial case, objectivity of detection was tested by investigating if the two experts actually did detect the same problems. First, the two experts independently analysed 30 dialogues. Each detected violation was then discussed in detail and a typology of violations established. This, highly task dependent, typology provides an overview of the different ways in which each individual guideline was violated in the corpus. The typology is useful for revising the dialogue model. The number of *individual* violations may support estimates of system performance and acceptability but is of little importance otherwise, as many violations are identical. All agreed violations could be classified under 24 different types. Of these, 15 were found by both experts whereas 9 types were found by either A1 or A2. Upon closer analysis, the cases belonging to 6 of the 9 complementary types turned out to be part of complex violations which had been discovered by both experts. The remaining 3 types only covered 1 case each.

Secondly, having discussed and classified 30 dialogues, the experts analysed another 15 dialogues from the Sundial corpus using the corpus dependent typology established during analysis of the first 30 dialogues. This facilitated dialogue annotation which could be reduced to references to a growing table of types. Slightly more type identities (17) were found in these dialogues but also slightly more type complementarities (12). However, all cases belonging to 8 of the 12 complementary types were part of complex violations that had been discovered by both experts. The remaining 4 types only covered one case each.

Results on objective (complex) problem identification are thus encouraging. Still, improvements in objective type identification would be desirable. At least two issues will have to be addressed in order to solve this problem: the concept of (corpus dependent) “types” needs elaboration and we have to construct more thorough explanations of each guideline and it use.

Transferability of the tool

However general the tool turns out to be eventually, it remains of little utility until other developers are able to use it with modest training and without requiring the presence or constant advice of its originators. In an early test of tool transferability, we trained a visiting researcher (A3) in how to use the tool. By way of introduction, A3 received the cooperativity guidelines (Figure 1), a paper on their background and development including examples of guideline violations, and a detailed tool application walkthrough of three Sundial dialogues. The complete analysis of one of these dialogues was given to him on paper. Having independently analysed a first set of 15 dialogues, A3 asked for, and had, a joint walkthrough of one of those. A3 received no detailed written information on how to use the guidelines.

We analysed the correspondence between the findings of the two experts and those of the novice in the first 30 dialogues. A3 found a total of 154 cases and 14 types, i.e. 80% of the average number of cases found by A1 and A2, and 72% of the average number of types found by A1 and A2. A3 found 10, or 42%, of the 24 types found by A1 and A2, and he found 4 new types. Three of these were part of complex violations that already had been observed by A1 and/or A2. The last type which covered only one case was not found by the two experts. Of the 154 cases found by A3, 26 cases were rejected, disagreed with or considered undecidable by A1 and A2. This should be compared to an average of 20 such cases found by the two experts.

Taking into account that A3 never received any formal instructions on how to use the guidelines but had to generalise from examples, his performance would seem acceptable. We now have to find out how to improve it further. The next step will be to introduce A3 to the use of corpus dependent violation typologies and then have A1, A2 and A3 analyse 10 dialogues from the remaining Sundial corpus. If the performance of A3 improves to the extent that transferability has been successful, we have to formalise what A3 needed to learn, thereby defining an explicit and simple training scheme for how to become an expert in using the tool without assuming person-to-person tuition. If this problem can be solved, the tool would have taken a significant step towards transferability.

Conclusion and future work

We find results so far encouraging. The tool has generalised well with respect to the Sundial corpus. A high degree of objectivity has been demonstrated with respect to the identification of complex dialogue design problems. Somewhat less objectivity was found in corpus dependent type identification. As regards transferability, the results obtained seem reasonable given the nature and amount of introductory material provided to A3. However, we clearly need a more systematic and elaborate way of demonstrating and explaining the use of each individual guideline. We hope this will also help the experts improve their corpus dependent type identification.

In DISC, we will continue testing the generality of the guidelines on a number of corpora and systems development processes. The planned next step is to test the guidelines on a sub-corpus of the Philips corpus which comprises 13.500 field test dialogues on train timetable information (Aust et al. 1995). This will add a new dialogue type, a new task type, and the circumstances of a field trial to the generality test of the tool. The problem of transferability will also be addressed in DISC. We need to provide explanations of how to use each individual guideline and to establish an elaborate set of examples illustrating the use of each guideline. Work has recently started on the development of a web-based tool that can explain and exemplify the guidelines. This tool will form part of future transferability tests.

References

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

Aust, H., Oerder, M., Seide, F. and Steinbiss, V.: The Philips Automatic Train Timetable Information System. *Speech Communication* 17, 249-262, 1995.

Aust, H. and Oerder M.: Dialogue Control in Automatic Inquiry Systems. *Proceedings of the ESCA Workshop on Spoken Dialogue Systems*, 121-124, Aalborg, Denmark, 1995. Also in: *Proceedings of TWLT9,* 45-49, Enschede, The Netherlands.

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: *Designing Interactive Speech Systems. From First Ideas to User Testing.* To be published by Springer Verlag, 1997.

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: What should your speech system say to its users, and how? Guidelines for the design of spoken language dialogue systems. To appear in *IEEE Computer*, 1997.

Blyth, B. and Piper, H.: Speech recognition: a new dimension in survey research. *Journal of the Market Research Society* 36(3), 1994, 183-203.

Bossemeyer, R.W. and Schwab E.C.: Automated alternate billing services at Ameritech: speech recognition and the human interface. *Speech Technology Magazin*e 5(3), 1991, 24-30.

DARPA. *Proceedings of the Speech and Natural Language Workshop.* San Mateo, CA: Morgan Kaufmann, 1992.

Dybkjær, H., Dybkjær, L. and Bernsen, N.O.: Design, formalisation and evaluation of spoken language dialogue. *Proceedings of the TWLT9 Workshop*, Enschede, 1995, 67-82.

Franco, V.: Automation of operator services at AT&T. *Proceedings of Voice'93,* San Diego, 1993.

Fraser, N.M. and Thornton, J.H.S.: VOCALIST: a robust, portable spoken language dialogue system for telephone applications. *Eurospeech'95,* Madrid, 1995, 1947-50.

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

Grice, P.: Logic and conversation. In Cole, P. and Morgan, J.L., Eds. *Syntax and Semantics,* Vol. 3, *Speech Acts,* New York, Academic Press, 41-58, 1975.

Lamel, L., Bennacef, S., Bonneau-Maynard, H., Rosset, S. and Gauvain, J.L.: Recent developments in spoken language systems for information retrieval, *Proceedings of the ESCA Workshop on Spoken Dialogue Systems,* Vigsø, Denmark, 1995, 17-20.

Ortel, W.C.G.: Observed long-term changes in customer calling patterns in a telephone application using automatic speech recognition, *Proceedings of Eurospeech '95,* Madrid, 1995, 269-272.

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

Peckham, J. and Fraser, N.M.: Spoken language dialogue over the telephone. In H. Niemann, R. de Mori and G. Hanrieder (Eds.): *Progress and Prospects of Speech Research and Technology.* Sankt Augustin: Infix, 1994, 192-203.

Peckham, J. and Fraser, N.M.: *Speech Understanding and Dialogue.* Cambridge, MA: MIT Press, (forthcoming).

Wahlster, W.: Verbmobil - Translation of Face to Face Dialogues. Machine Translation Summary IV, Kobe, 1993.