Different Spoken Language Dialogues for Different Tasks

A Task-Oriented Dialogue Theory

Laila Dybkjær, Niels Ole Bernsen and Hans Dybkjær
Centre for Cognitive Science, Roskilde University
PO Box 260, DK-4000 Roskilde, Denmark

Abstract

Spoken language dialogue is a comfortable form of communication between humans and computers, which is present in a growing number of commercial systems. For each task which can be comfortably performed in spoken language dialogue with the computer, there is an equivalence class of tasks which can be performed using similar dialogue management technology. Each such task class has a number of minimum functional requirements which, once they have been met by the technology, will enable comfortable spoken language human-computer dialogue. The paper presents these requirements in terms of dialogue elements such as initiative, system feedback, predictions and system focus, dialogue history, user models and meta-communication. Three increasingly complex task class/dialogue type pairs are distinguished and their corresponding minimum dialogue elements are presented and illustrated from our own development of spoken language dialogue systems. The result is a first version of task-oriented dialogue theory which may support the design and specification of increasingly sophisticated spoken language dialogue systems.

1. Introduction

Spoken language dialogue is an inherently habitable (or comfortable) form of communication between humans. It is spontaneous, informal and mastered by virtually everyone [13]. Spoken language is therefore desirable as a modality in human-computer communication, whether this modality be used alone, such as in spoken dialogue over the telephone, or in multimodal combination with other modalities, such as graphics. Generic research and development in spoken language dialogue systems (SLDSs) aim to augment the dialogue understanding technologies of systems, and hence their human-computer interfaces, through

improved understanding of spoken language input, improved spoken language generation and improved dialogue management. Each incremental step in this direction of improved dialogue understanding technologies is likely to simultaneously increase the interface capacities of multimodal systems involving speech as one of their modalities. 'Capacity' may in this context be measured in terms of types of tasks. I.e., the larger the capacity of a certain interface modality, such as speech, or of a certain multimodal interface combination, such as speech and CAD graphics, the more task types can be habitably supported by the modality or modalities. For the purpose of this paper, an SLDS is defined as a system which has (input) speech understanding, thus excluding, e.g., 'speech typewriters' (no understanding) [18] and standard voice response systems (no spoken input).

Several kinds of SLDS, some of which are commercially available, today satisfy conditions of minimum habitability, i.e. their task performance is minimally acceptable to users. Other SLDSs, however, are not yet minimally habitable as their dialogue understanding technologies are still too deficient. In this chapter we want to look at how to improve SLDS habitability through improvements in dialogue management. The habitability of SLDSs depends heavily on the dialogue model which is the active and controlling part of an SLDS, defines much of the user interface and functionality of such systems, and which may also support speech recognition and language processing through prediction of user input. The dialogue model must be designed on the basis of an analysis of the tasks to be interactively carried out by user and system. This is why the theory to be presented below may be characterised as a task-oriented dialogue theory.

The central concept in dialogue design is that of a task. The task type for which the system is to be built, determines which type of user-system dialogue is needed to achieve minimum habitability. If some specific task can be managed by an SLDS in a way which is minimally habitable, there will be an entire category of broadly equivalent tasks which can be managed in a similar way. Tasks which can be habitably managed by SLDSs currently range from small and simple tasks performed in single-word dialogue, through to larger, well-structured tasks accomplished in real dialogue turn-taking but allowing little or no user dialogue initiative. Current commercial SLDSs are all based on single-word dialogue whereas SLDSs having system-directed dialogue are coming close to commercialisation. The major research challenge today is the management of mixed initiative dialogue. Progress in the management of increasingly difficult task types by means of increasingly complex dialogue has the additional effect of improving, beyond minimum habitability, dialogue performance on less demanding task types.

The task-oriented dialogue theory to be presented below is based on, and will be illustrated from, work on SLDS prototypes in the Danish project on spoken language dialogue systems. The Danish Dialogue project is a collaboration between the Center for PersonKommunikation (CPK), Aalborg University, the Centre for Language Technology (CST), Copenhagen, and the Centre for Cognitive Science (CCS), Roskilde University. The aim is to develop two

application-oriented, real-time, speaker-independent SLDS prototypes called P1 and P2 in the domain of Danish domestic airline ticket reservation and flight information accessed through the telephone. We have developed the P1 dialogue using the Wizard of Oz method and a corpus of human-human dialogues in the task domain [1, 9]; implemented the P1 dialogue [8]; and are presently testing the system.

P1 allows users to interactively perform the ticket reservation task which is a large, well-structured task and well suited for system-directed dialogue. P1 takes as input a speech signal which is recognised using Hidden Markov Models and passed as a sentence hypothesis to the linguistic analysis module. This module uses a chart parser to perform a syntactic and semantic analysis of the sentence and represents the result in a set of frame-like structures called semantic objects. The dialogue handling module performs a task-oriented interpretation of the semantic objects received from the linguistic analysis module and takes action according to this input, e.g. through updates or queries to the application database or decisions on the next output to the user. In P1 the output module uses pre-recorded speech rather than language generation and text-to-speech synthesis.

The task to be addressed in P2 will be flight information inquiry which is an ill-structured task that does not lend itself to fully system-directed dialogue. P2 will incorporate more advanced technological solutions than P1, such as superior output functionality based on language generation and speech synthesis. Improved recognition techniques, an improved parser and extended grammars and vocabulary will allow the design of more habitable dialogues than in P1. P2 is currently being specified based on P1 and the dialogue theory to be presented shortly.

In what follows, we first discuss the decomposition of dialogue tasks (section 2) and present a division of tasks into increasingly complex types which require increasingly complex dialogue types to enable habitable user-system interaction (section 3). Depending on its complexity, each dialogue type has to incorporate a certain number of *dialogue elements* in order to satisfy conditions of minimum habitability. The dialogue elements, i.e. initiative, system feedback, predictions and system focus, dialogue history, user models and meta-communication, are discussed in sections 4 to 9 and summarised in section 10. Section 11 concludes the chapter.

2. Dialogue Level Decomposition

There is broad agreement in the literature on the number of hierarchical levels into which task-oriented dialogues should be decomposed for the purpose of adequate description [cf. 4]. Task-oriented SLDS dialogue may be decomposed into the following three levels of description and analysis each of which is illustrated by examples from P1:

1. *Task level.* A *dialogue task* N consists of one or more tasks which are referred to as *dialogue sub-tasks* relative to N. Tasks may be embedded in, and

hence be sub-tasks relative to, other tasks. Task N is realised through realising its dialogue sub-tasks a, b, c, ..., n. The global unfolded dialogue task structure shows all tasks and their embeddings, i.e. which tasks are sub-tasks relative to a given task. The global unfolded dialogue task structure of P1 illustrating all tasks and their embeddings is shown schematically in figure 1. Only the structure of the domain-related dialogue is shown, not the meta-communication (cf. section 9).

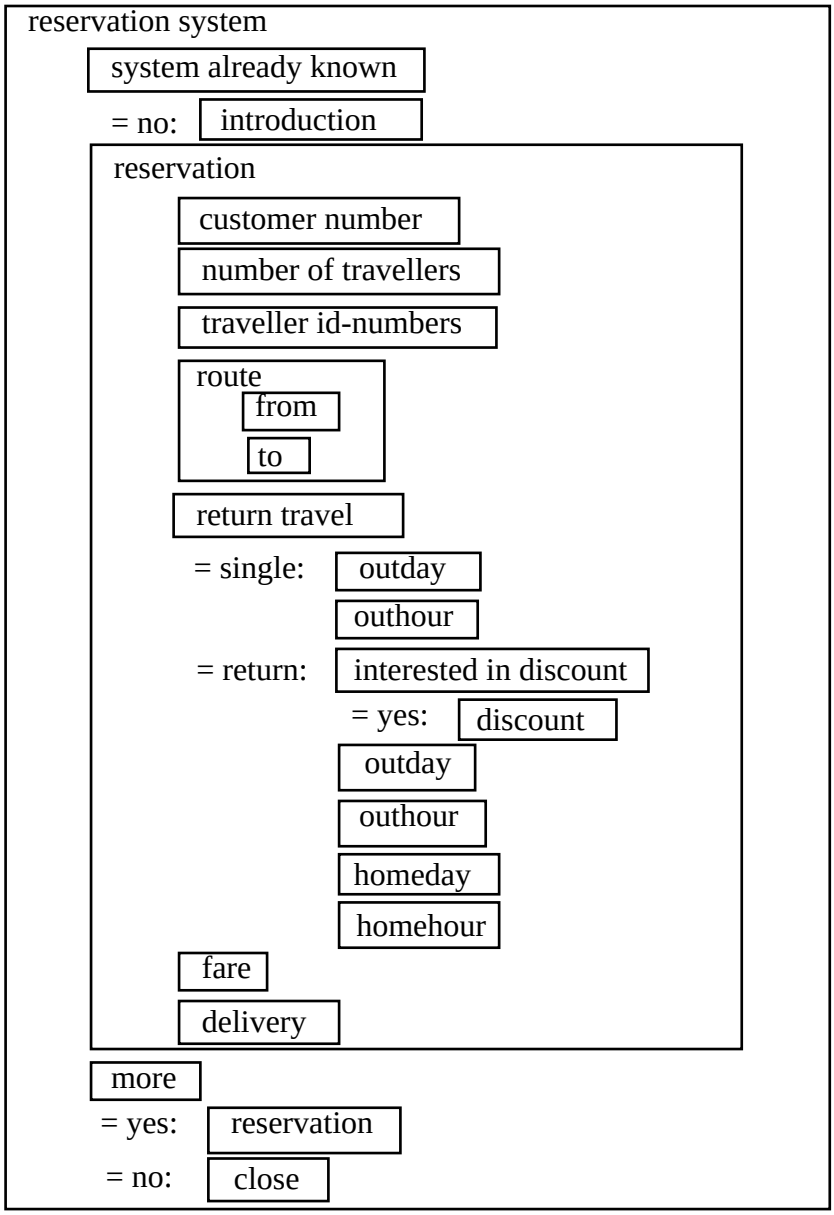

Figure 1: The unfolded domain dialogue task structure for P1. A labelled box indicates a task. If a box A contains another box B then B is a sub-task relative to A. At some points during dialogue the structure to follow depends on the user's answer to the most recent question. In such cases an answer is indicated as '= [answer]:'

followed by the tasks to be performed. The dialogue task structure is a cyclic graph with conditional branches.

2. *Turn-taking or utterance level.* In SLDSs task N is realised through user-system turn-taking involving a sequence of *dialogue turns* S (System)1, U (User)1, S2, U2, ..., Sn, Un. A turn consists of a user or system *utterance*. Each turn can at least be characterised by the dialogue act(s) it contains and by whether the speaker (user or system) *has the initiative* or *responds* to an initiative taken by the interlocutor ([4], section 4 below). The following example shows the completion of three sub-tasks during six dialogue turns:

S1: How many persons are going to travel?
U1: One.
S2: What is the id-number of the person?
U2: Fifty-seven.
S3: Id-number fifty-seven Jens Hansen. Where does the travel start?
U3: In Aalborg.

3. *Dialogue act level.* An utterance may contain one or more *dialogue acts*. In the example above, the system's third turn S3 contains two dialogue acts, the first being an assertion (a declarative act stating a fact) which provides echo feedback on the dialogue act in the preceeding user turn U2, the second being a question to the user. Dialogue acts are similar to speech acts [16]. Dialogue acts are dynamic semantic entities, i.e. they occur in a specific dialogue task context and are defined in terms of their modification of that context [5].

3. A Task Type Taxonomy Based on Task Structure and Task Size

Many complex tasks, such as the flight ticket reservation task of P1, have a stereotypical structure. A *task stereotype* prescribes which information must be exchanged between the dialogue partners to complete the task and, possibly, in which order this may be done naturally. The dialogue task structure of figure 1 expresses the reservation task stereotype. This structure conforms to the most common structure found in corresponding human-human reservation task dialogues recorded in a travel agency [9].

The work on P1 has suggested that, when shared by user and machine, task stereotypes strongly facilitate dialogue systems design because they allow the computer to direct the dialogue through asking questions of the user without the user feeling this to be a major drawback of the design. Adding observations by [19], the following hypothesis emerges: *System-directed dialogue is acceptable to users in cases where there is a directly applicable task stereotype which is assumed by the user to form part of the system's expertise.* Thus, system and user do not even have to share stereotypical task knowledge in cases where (1) the system has sufficient knowledge of the user's situation to embark on the stereo-

typical task right away and (2) the user has sufficient confidence in the system's task knowledge to let it do so. This would bring a considerable number of task types within the scope of system-directed spoken language dialogue, including many tasks in which the user is novice or apprentice and the system acts as an expert instructing the user on what to do or say.

In addition to the flight ticket reservation task, a travel information task including flight schedules, fares and travel conditions was specified for P1 using the Wizard of Oz technique. The information task was not implemented, however, and will only be so in P2, for the following reason. Whereas ticket reservation tasks conform to a single, basic stereotype, travel information tasks do not. Knowing that a user wants travel information does not help the system know what to offer and in which order. This means that travel information tasks are not well suited for system-directed dialogue. The corresponding hypothesis is that *if a task has no stereotypical structure but contains a large number of optional sub-tasks, then the system cannot take and preserve the initiative during dialogue without unacceptable loss of dialogue naturalness.* In such cases, mixed initiative dialogue is necessarily called for to allow an acceptable minimum of habitability. In the task stereotype case, although always preferable to rigid, system-directed dialogue, mixed initiative dialogue is not strictly required. The class of non-stereotypical tasks seems to be quite large including, i.a., tasks in which users seek information, advice, or support, or otherwise want to selectively benefit from a system's pool of knowledge or expertise.

Not only task structure but also task size contributes to determining the complexity of the dialogue understanding technology needed to habitably manage a task. The above distinction between well-structured and ill-structured tasks and the dialogue types required by each, is valid for larger tasks. Small and simple tasks, on the other hand, whether well-structured or not, are less demanding. The distinctions between smaller and larger tasks, well-structured and ill-structured tasks, and the minimum demands which each task type imposes on dialogue type and other dialogue understanding technology, are shown in figure 2.

Minimum requirements are such which must be satisfied in order to build a minimally habitable system for a task of a certain type. The division into three task types is rough but illustrates the state of today's technology and the points on which improvements are needed. The technology is close to being available for developing commercial products capable of managing the task types described in the first two columns of figure 2, and the technology is approaching the prototyping stage with respect to the task type of the third column of figure 2. Just as importantly, every time dialogue understanding technology has improved enough for the management of a new task type to be possible, significant improvements in the habitability of SLDSs for already mastered task types become possible. Thus, in figure 2, individual elements from a succeeding column may be used to improve the dialogue performance of systems belonging to preceeding columns.

Small and simple tasks can be managed in a dialogue based on single-word user utterances. Multi-user systems such as [14] typically have small speaker-in-

dependent vocabularies whereas systems meant for personal use often have a somewhat larger, speaker-adaptive vocabulary [6]. There is a significant difference in dialogue complexity between single-word dialogue, on the one hand, and system-directed dialogue and mixed initiative dialogue on the other. The difference primarily derives from increased user utterance length.

Single-word utterances are simple to process. In some single-word dialogue systems the initiative may lie entirely with the user. In such cases the dialogue structure is often flat as in the example in [6] of a system permitting oral manipulation of files, etc. on a PC. Other single-word dialogue systems are fully system-directed, such as [14]. Typically, their dialogue structure is shallow but not totally flat. System feedback is relevant in single-word dialogue in cases where it is crucial that the user's intention has been correctly understood before the task is carried out, e.g. when transferring money to another account.

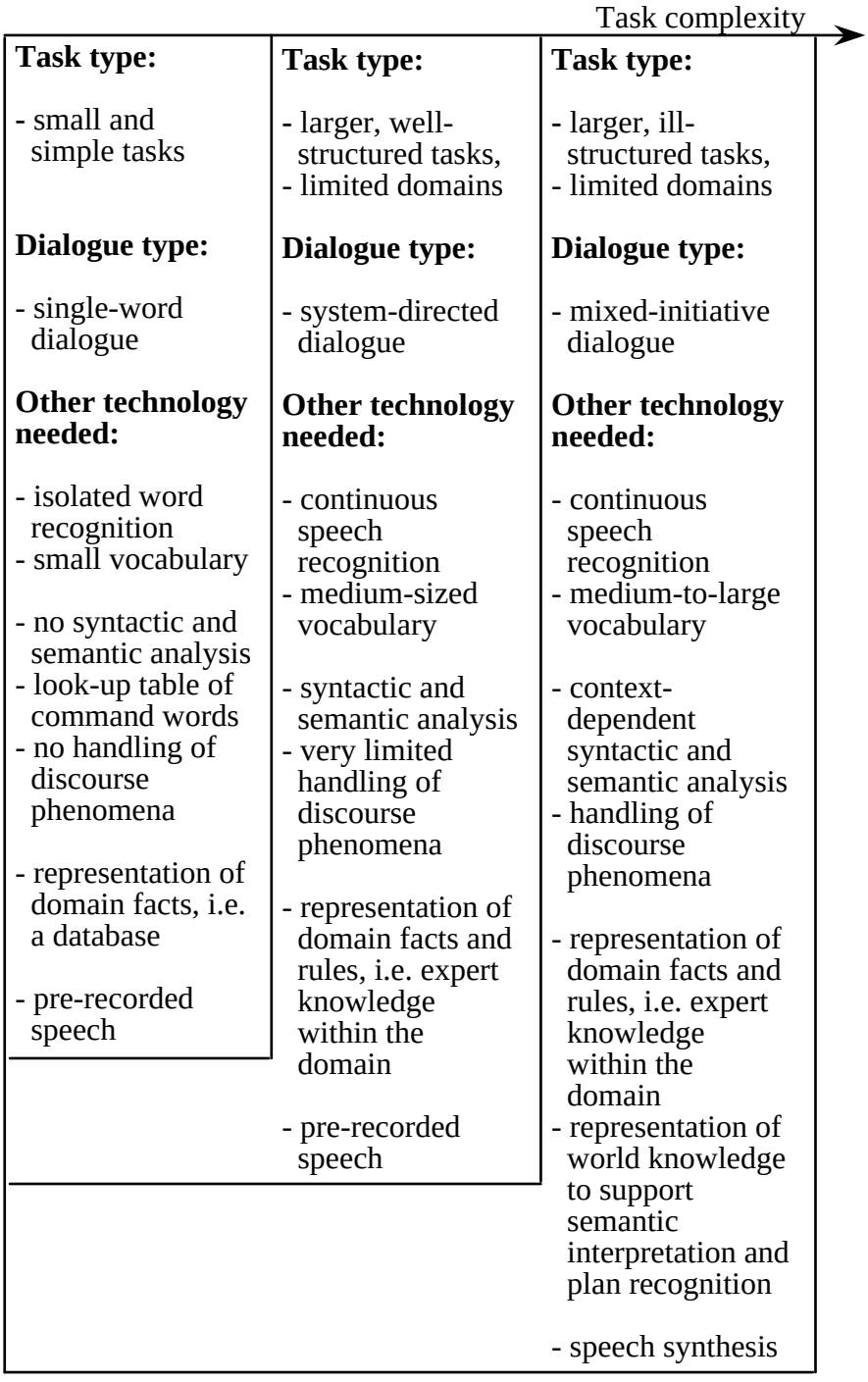

Figure 2: In order to manage a particular task type in a way which is acceptable to users, a number of minimum requirements on the dialogue understanding technology must be met. These include, firstly, the type of dialogue needed. Secondly, the dialogue type in its turn defines requirements on speech recognition, linguistic analysis, domain representation, and output facilities.

We call elements such as initiative and system feedback *dialogue elements*. In single-word dialogue no dialogue elements in addition to those mentioned above are strictly needed, cf. [2]. In the remainder of this paper, we focus on the dialogue elements needed to habitably perform larger tasks and more complex dialogues.

4. Dialogue Initiative

The interlocutor who controls the dialogue at a certain point has the *initiative* at this point and may decide what to talk about next, such as asking questions which the dialogue partner is expected to answer. As only the stereotypically structured reservation task has been implemented in P1, it seems acceptable that the system, with two exceptions to be mentioned shortly, takes and preserves the initiative throughout the dialogue. The distinction between user and system initiative, therefore, has not been explicitly represented in the implementation. The system takes and preserves the initiative by concluding all its turns (except when closing the dialogue) by a question to the user. The questions serve to implicitly indicate that initiative belongs to the system rather than the user. Only in meta-communication is the user allowed to take the initiative by using keywords (cf. section 9) which enable the system to immediately identify both the user initiative and the task the user intends to perform.

Even if the described solution may work for stereotypical tasks, keywords-to-be-remembered are unnatural and systems for non-stereotypical tasks need user initiative. P2 will have mixed initiative dialogue for improved naturalness of meta-communication and in order to solve the problem posed by the non-stereotypical information task (cf. section 3). If an explicit system representation of who has the initiative throughout a dialogue will be needed in order to achieve those aims, one way for the system to establish who has or takes the initiative, might be to use control rules based on dialogue context and a simple taxonomy of user dialogue acts [19].

The correlated distinctions between stereotypical tasks/system initiative and unstructured tasks/mixed initiative dialogue provides a rough guideline for determining where the emphasis should lie given a certain type of task to be performed interactively between user and system. In fact, there seems to be a continuum between full system control through use of questions, declarative statements or commands, and mixed initiative dialogue in which the system only assumes control when this is natural. Even SLDSs for stereotypical tasks need some measure of mixed initiative dialogue to be fully natural [15, 17]. And systems performing non-stereotypical tasks, such as large numbers of unrelated sub-tasks, are often able to go into system-directed mode once a stereotypically structured sub-task which the user wants performed, has been identified [11].

5. System Feedback

In the context of SLDSs, system feedback is a repetition by the system of key information provided by the user. The provision of sufficient feedback to users on their interactions with the system is particularly crucial in speaker-independent SLDSs because of the frequent occurrence of misunderstandings of user input. The user needs to know whether or not a task has been successfully completed and hence whether repair or clarification is needed.

P1 provides *continuous feedback* on the user commitments made during a task. When the system decides that it has sufficient information to complete a sub-task, the user receives feedback on that information. Users who accept the feedback information do not have to reconfirm their commitment as the system will carry on with the next sub-task in the same utterance. Two such cases of feedback can be seen in the example dialogue in section 2 above: In S2, the term 'person' confirms that only one person will be travelling. This we will call *masked echo* feedback. In S3, the id-number provided by the user is repeated and the name of the person added for extra confirmation. This we will call *echo* feedback. Masked echo and echo feedback are obviously more parsimonious than, and hence often preferable to, *explicit* feedback which requires the system to repeat what the user just said with an added request for confirmation from the user. A sophisticated solution may be to use acoustic scores, or acoustic scores combined with perplexity as a basis for determining which type of feedback to give to the user in a particular case, as proposed by [4]. If the score drops below a certain threshold indicating considerable uncertainty about the input, explicit feedback might be offered. If the user does not accept the feedback information, meta-communication is needed (cf. section 9).

In addition to continuous feedback, P1 offers *summarising feedback.* On closing the reservation task, the system summarises the information provided by the user. Summarising feedback provides the user with an overview of the commitments made and thus has a role which is distinctly different from that of continuous feedback. Users should be able to initiate meta-communication in cases where the summarised commitments are no longer viewed as satisfactory.

The types of feedback already mentioned will probably be sufficient for mixed initiative dialogue.

6. Prediction and System Focus

Predictions are expectations as to what the user will say next and help identify the sub-vocabulary and sub-grammars to be used by the recogniser. Predictions constrain the search space and express the sub-tasks which the user is expected to address in the next utterance. If the user chooses to address other sub-tasks, system understanding will fail unless some prediction-relaxation strategy has been adopted. The more stereotypical structure a task has, the easier it is to make good predictions provided the user is cooperative. One key reason why practical mixed initiative systems are hard to realise is that they make user input prediction more difficult, especially in non-stereotypical tasks [11]. In mixed

initiative dialogue in general, and in non-stereotypical task dialogue in particular, the first challenge the system faces on receiving a user utterance, is to identify the sub-task the user intends to perform.

Predictions are based on the set of sub-tasks currently in *system focus.* The set of sub-tasks in system focus are the tasks which the user is allowed to refer to in the next utterance. A useful heuristics for stereotypical task systems seems to be that the set of sub-tasks in system focus always include the preceeding sub-task (if any), the current sub-task, the possible succeeding sub-task(s), according to the default dialogue task structure, and the meta-communicative tasks which might be initiated by the user. Ideally, the system focus should correspond to the *common dialogue focus* shared by the interlocutors. The heuristics just mentioned should make the correpondence achievable in many types of task-oriented dialogue based on task stereotypes, provided that the needed prediction sets are technologically feasible. In such cases, the heuristics may ensure correspondence between system focus and the set of sub-tasks which the user will find it natural to address at a given point during dialogue. In general, of course, the more overlap there is between system focus and user focus, the more likely it is that the dialogue will proceed smoothly. This field is one in which practical systems design expects to benefit from basic research on discourse.

In P1, the dialogue handler predicts the next possible user utterances and tells the speech recogniser and the parser to download the relevant sub-vocabulary and sub-grammars. To obtain both real-time performance and acceptable recognition accuracy it has been necessary to restrict sub-vocabularies to contain at most 100 words [7]. The system's predictions include the current sub-task and the meta-communicative possibilities of the user saying 'correct' or 'repeat'. In some cases P1's predictions include more than the current sub-task. For instance, when the system expects an arrival airport, the departure airport is also included in its predictions and may therefore be provided by the user in the same turn as the arrival airport.

Information on the sub-tasks in system focus is hardwired in P1. For each point in the dialogue structure it has been decided which sub-grammars should be active and how the system's utterances should be expressed. The decision on sub-grammars depends on the number of active words required. This approach will not work for mixed initiative dialogue where the user has the opportunity to change task context (or topic) by taking the initiative. When part of the initiative is left to the user, deviations from the default domain task structure may be expected to occur from time to time and in such situations the system has to be able to determine the set of sub-tasks in system focus at run-time. Mixed initiative dialogue therefore requires a dynamically determined set of sub-tasks in system focus.

7. Dialogue History

A *dialogue history* is a log of information which has been exchanged so far in

the dialogue. We distinguish between four different kinds of dialogue history each of which has its own specific purpose. Further distinctions among dialogue histories are likely to be needed at some stage [cf. 10]. Firstly, the *linguistic dialogue history* logs the surface language of the exchanges (i.e. the exact wording) and the order in which it occurred. Linguistic dialogue history is primarily used to support the resolution of anaphora and ellipses and has to do its work before producing semantic frames for the dialogue handler. Therefore linguistic dialogue history has a closer relation to the linguistic module than to the dialogue model. It is an open question if SLDSs will ever need access to the entire linguistic dialogue history or whether a window of, say, the four most recent user-system turns is sufficient. P1 does not need a linguistic dialogue history because it only accepts a maximum average user utterance length of 4 words. With P2's longer user utterances and increased user initiative, a linguistic dialogue history will be necessary to allow, i.a, anaphora resolution.

User input surface language is not needed for dialogue management which only requires representation of input order and semantics. We call a history which records the order of dialogue acts and their semantic contents a *dialogue act history*. In P1 the dialogue act history logs

1. an identification of the previous sub-task in order to be able to make corrections on user request;
2. the logical contents of the latest system question. It is, e.g., important to the interpretation of a yes/no answer from the user to know how the question was phrased. It makes a difference if the question was "Is it a one-way travel?" or "Is it a return travel?";
3. the semantic contents of the user's latest utterance.

The dialogue act history is used for correcting the most recent user input. Corrections to information exchanged prior to the most recent user input cannot be made in P1. A larger dialogue act history would probably not help in this case. As in human-human dialogue, the most convenient solution for the user will be to explicitly indicate the piece of information to be corrected. This requires, i.a., a task record (see below), maintenance of inter-dependencies between task values and an implementational strategy for revisiting earlier parts of the dialogue structure.

The third kind of dialogue history is the *task record* which logs task-relevant information that has been exchanged during a dialogue, either all of it or that coming from the user or the system, depending on the application. All task-oriented dialogue systems would seem to need a task record because they have to keep track of task progress during dialogue. However, a task record does not keep track of the order in which information has been exchanged and ignores insignificant exchanges relative to the task. The task record also logs which tasks are pending and which ones have been completed. The system may have to suspend the current task if it discovers that it needs some value in order to proceed, which can only be obtained by performing a task which is prior in terms of the task structure. For instance, to determine whether a certain departure hour is acceptable it is necessary to know the date of departure.

In P1 all values obtained from the user concerning the reservation and the

extent to which the values have been checked by the system, are recorded. Pending sub-tasks are not allowed.

The fourth kind of dialogue history is the *performance record.* This record updates a model of how well the dialogue with the user proceeds and may be used to influence the way the system addresses the user. P1 does not have a user model-based performance record. The next section discusses user modelling in more detail.

8. User Modelling

In human-human dialogue, a participant is normally prepared to change the way the dialogue is being conducted in response to special needs of the interlocutor. During dialogue each participant builds a model of the interlocutor to guide adaptation of dialogue behaviour (cf. the performance record, section 7). In other cases, a participant already has a model of the interlocutor prior to the dialogue, upon which to base dialogue behaviour. The participant knows, for instance, that the interlocutor is a domain expert who only needs update information. A reservation system might do the same by, e.g., using the user's previous ticket reservation record as a guide to how to handle the dialogue - or by simply asking the user.

P1 incorporates a small amount of user modelling. In the dialogue opening task phase, the user is asked: "Do you know this system?" (cf. figure 1). If the answer is "No", the user is presented with an introduction on how to use the system. If the answer is "Yes", the introduction is by-passed. In P2, we will try to extend system adaptivity by introducing a performance record which helps the system determine how to address the user, i.e. whether, for instance, increased use of spelling requests, explicit yes/no questions or multiple choice questions might be helpful to allow the dialogue to succeed. Otherwise, the sky is the limit in how adaptive user models may be created and used in future generations of SLDSs.

9. Meta-Communication

Meta-communication is distinct from *domain communication* and serves as a means of resolving misunderstandings and lacks in understanding between the dialogue partners during dialogue. Today's SLDSs require that users provide cooperative utterances so that it is possible to make valid predictions [3, 10]. Cooperative utterances are utterances which a user has a right to expect the system to be able to understand. It is up to the system to inform users on the system's understanding capabilities and limitations. Cooperative utterances must conform to this information. However, even when users are cooperative the system may fail to understand them, or misunderstand them. In current SLDSs, meta-communication for *dialogue repair* is essential because of the sub-optimal quality of the systems' recognition of spontaneous spoken language.

Similarly, meta-communication for *dialogue clarification* is common in human-human dialogue and serves to resolve cases of ambiguous or incomplete information, and the ability to perform clarification dialogues is generally needed in SLDSs. We shall look at dialogue repair in what follows.

If understanding failure is due to difficulties in recognising a user's pronunciation of certain words, a first reaction could be to ask the user to repeat the utterance. This is the least possible step in the direction of trying to repair an understanding problem. However, if understanding failure is due to, e.g., an overly complicated utterance, simple repetition will not help. In this case it is necessary to make the user express the information more simply. As it is probably impossible for the system to always detect exactly why understanding has failed, a general method for repairing system understanding problems is needed. System-prompted graceful degradation appears to be a promising approach (for a combination of graceful degradation and feedback, see [12]).

When using graceful degradation, the system will explicitly ask the user to provide the missing information in increasingly simple terms. This degradation in *user input level* will continue until either the system has understood the input or no further degradation is possible. In P2, distinction will be made between the following five different, system-prompted user input levels, roughly listed in the order of increasing input complexity: (1) spelling question, (2) yes/no question, (3) multiple choice question, (4) focused question and (5) open, mixed initiative. It is not always a solution to degrade to the level immediately below the current one. For instance, when asking for an arrival airport it would not make sense to use a multiple choice question if there were, e.g., ten possible destinations. In this case the next relevant level would be to ask the user to spell. So the problem is how to decide the next relevant level. This may be done as follows: For each piece of information to be obtained, all five levels are indicated together with a grammar telling how to ask for that information. If it does not make sense to use a certain level, no grammar is indicated, and if it only sometimes makes sense the grammar is conditioned. When the system has understood the user, the dialogue returns to the user input level used immediately before degradation. A tentative method for carrying out graceful degradation when system understanding fails, involves the following three steps:

1. Initialisation: If the system does not yet have the initiative it takes the initiative and asks the user a question concerning what it believes to be the topic of the user's utterance. The topic may be determined on the basis of the system focus set. If understanding fails again the system proceeds to step 2, otherwise degradation stops. For instance, the system believes that the user wants to make a reservation but has no information on the reservation yet, and therefore asks "Where does the travel start?".

2. Either repeat or do explicitness iteration: In explicitness iteration, the system makes explicit to the user what was implicit in its original question. If understanding still fails, the system proceeds to step 3, otherwise degradation stops. The question "Where does the travel start?" can be repeated but hardly made much more explicit (e.g. "From which airport does the travel start?"). An

example of explicitness iteration is when the system stresses that the user's answer should mention one of three options offered by the system.

3. Level iteration: The system asks an equivalent question which can be answered in a different and simpler way, i.e. degrades to the next relevant level and then proceeds to step 2 if understanding still fails, otherwise degradation stops. When no lower level exists a bottom stop condition is activated, such as asking the user to address a human travel agent. In the "Where does the travel start?" case, for instance, the system asks the user to spell the name of the departure airport.

P1 does not offer graceful degradation. P1 initiates repair meta-communication by telling the user that it did not understand what was said or, in case the user signals, using the *correct* command, that the system has misunderstood something, by repeating its penultimate question.

We would like to end this section by pointing out that, in practical applications, the term 'meta-communication' must be taken in a rather wide sense. Our P1 work suggests that, in addition to repair and clarification functionality, the following functions will be needed in practice:

- a 'wait' function for use when the user needs time, e.g. to think or to talk to somebody;
- a 'dialogue help' function for use when users need help from the system to get on with the dialogue. Actions such as unexpected, confusing or irrelevant answers or repeated use of the repeat function may indicate a need for help;
- a 'restart' function for use when the user needs to start all over again, e.g. because too many things have gone wrong during the dialogue.

However, such functions should not be added by introducing new keywords which users have to remember but rather by allowing mixed initiative dialogue for meta-communication purposes.

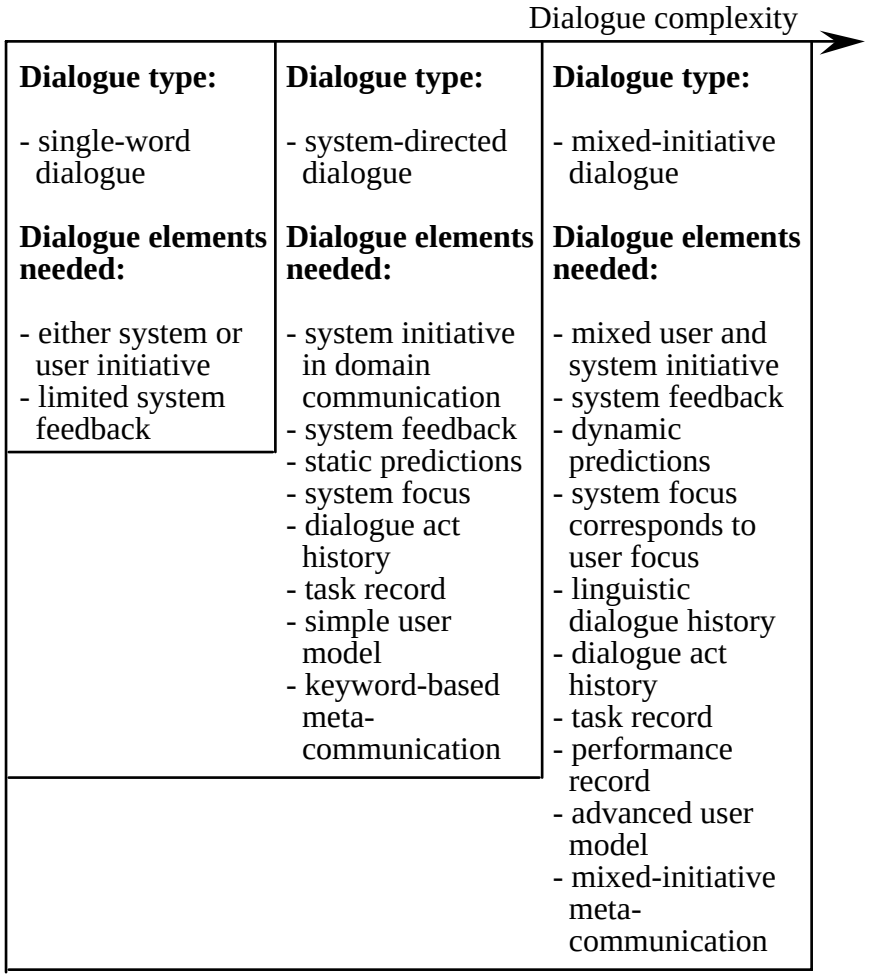

Figure 3: The more sophisticated the dialogue, the larger the demands on dialogue theory and the dialogue elements supporting the dialogue model.

10. Summary of Dialogue Elements

Figure 2 distinguished between three increasingly complex task/dialogue types. These task/dialogue types require an increasing number of dialogue elements to ensure habitability. The dialogue elements were discussed and illustrated by examples from each of the three task types while maintaining the focus on system-directed and mixed initiative dialogue. Figure 3 presents an overview of the dialogue elements which are needed as a minimum for each of the three dialogue types to support habitable dialogue.

11. Conclusion

As the development of SLDSs moves from science towards craftsmanship, and from the laboratory into commercial applications the need arises to address, in a

systematic and integrated fashion, the different aspects of the design process which must be mastered to build usable and habitable systems. Steps towards an incremental theory of SLDS functionality have been presented, amounting to a 'toolbox' of functionalities whose individual tools have been correlated with task class/dialogue type pairs of increasing complexity. There is no doubt that future dialogue management needs will give rise to finer distinctions among task classes than the one presented here, and that future systems will incorporate dialogue elements in addition to those described, thus adding structure and contents to task-oriented dialogue theory. However, it is already possible to cover large parts of the task space for which SLDS technology is appropriate through suitable specifications of the dialogue elements described above. It is hoped that the present version of task-oriented dialogue theory may be of use in the specification process.

References

1. Bernsen, N.O., Dybkjær, L. and Dybkjær, H.: Task-Oriented Spoken Human-Computer Dialogue. Report 6a, Spoken Language Dialogue Systems, CPK Aalborg University, CCS Roskilde University, CST Copenhagen, 1994.
2. Bernsen, N.O., Dybkjær, L. and Dybkjær, H.: A Dedicated Task-Oriented Dialogue Theory in Support of Spoken Language Dialogue Systems Design. In *Proceedings of ICSLP '94*, Yokohama, 18-22 September, 1994
3. Bilange, E.: A Task Independent Oral Dialogue Model. In *Proceedings of the 5th EACL*, Berlin, April, 83-88, 1991.
4. Bilange, E. and Magadur, J.-Y.: A Robust Approach for Handling Oral Dialogues. *Actes de COLING-92*, Nantes, August, 799-805, 1992.
5. Bunt, H.C.: Information dialogue as communicative action in relation to partner modelling and information processing. In Bouwhuis, D., Taylor, M. and Néel, F. (Eds.): *The Structure of Multimodal Dialogues Including Voice.* Amsterdam: North-Holland, 1-19, 1989.
6. Christ, B.: På talefod med PC'en. In *Alt om Data*, vol.3, pp.114-118, March 1992.
7. Dybkjær, H., Bernsen, N.O. and Dybkjær, L.: Wizard-of-Oz and the Trade-off between Naturalness and Recogniser Constraints. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, pp. 947-950, 1993.
8. Dybkjær, H. and Dybkjær, L.: Representation and Implementation of Spoken Dialogues. *Report 6b, Spoken Language Dialogue Systems, CPK Aalborg University, CCS Roskilde University, CST Copenhagen*, 1994.
9. Dybkjær, L. and Dybkjær, H.: Wizard of Oz Experiments in the Development of a Dialogue Model for P1. *Report 3, Spoken Language Dialogue Systems, STC Aalborg University, CCS Roskilde University, CST*

Copenhagen, 1993.

10. Eckert, W. and McGlashan, S.: Managing Spoken Dialogues for Information Services. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, pp. 1653-1656, 1993.
11. Guyomard, M., Siroux, J. and Cozannet, A.: The Role of Dialogue in Speech Recognition. The Case of the Yellow Pages System. *Proceedings of Eurospeech '91*, Genova, Italy, September, 1051-1054, 1991.
12. Heisterkamp, P.: Ambiguity and Uncertainty in Spoken Dialogue. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, pp. 1657-1660, 1993.
13. Lefebvre, P., Duncan, G. and Poirier, F.: Speaking with Computers: A Multimodal Approach. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, pp. 1665-1668, 1993.
14. MAX: Reference Card for MAX. ECHO, European Commission Host Organisation, B.P. 2373, L-1023 Luxembourg G.D., 1991.
15. Peckham, J.: A New Generation of Spoken Dialogue Systems: Results and Lessons from the SUNDIAL Project. In *Proceedings of Eurospeech '93*, Berlin 21-23 September, pp. 33-40, 1993.
16. Searle, J.R.: *Speech Acts*. Cambridge: Cambridge University Press, 1969.
17. Seneff, E., Hirschman, L. and Zue, V.W.: Interactive Problem Solving and Dialogue in the ATIS Domain. *Proceedings of the Pacific Grove Workshop*, CA, 354-359, February, 1991.
18. Takebayashi, Y., Tsuboi, H., Sadamoto, Y., Hashimoto, H. and Shinchi, H.: A Real-Time Speech Dialogue System Using Spontaneous Speech Understanding. In *Proceedings of ICSLP '92*, Banff, 12-16 October, pp. 651-654, 1992.
19. Walker, M. and Whittaker, S.: Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. *Proceedings of the ACL*, 70-79, 1990.

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