University of Leeds

SCHOOL OF COMPUTER STUDIES

RESEARCH REPORT SERIES

Report 97.06

Dialogue Management Systems: a Survey and Overview

by

Gavin E Churcher, Eric S Atwell, Clive Souter

February 1997

Contents

ABSTRACT

This report provides an overview of the current issues and techniques for the modelling of dialogues using a computer. A dialogue management system can manage a dialogue between two or more agents, be they human or computer. Recently, increasingly complex dialogues are being modelled which allow a range of discourse phenomena including ellipsis and anaphoric reference. Such dialogues are thought to be similar to those between two humans, and accurate modelling of these phenomena leads to "pleasant", i.e. easy to talk to, and natural human-computer dialogues. Dialogue management can be classified into three often overlapping approaches: discourse grammars, plan-based and collaborative approaches. The design of a system often begins by eliciting the language used initially between two humans and later by Wizard of Oz experiments. Special issues relating to dialogue management systems are discussed including recovery strategies from different types of errors and the coding of dialogue in corpora. Lastly, approaches to evaluation are briefly discussed from the qualitative and quantitative viewpoints, recognising the importance and size of this sub-field.

1. Introduction

This report looks at the relatively new field of dialogue management systems and provides an overview of the central issues and techniques used. Since the qualities of discourse were first comprehensively explored, see for example Levinson (1983) and Leech (1983), computationally viable techniques to model it have been gradually developed with increasing sophistication. Dialogue management systems allow a computer to model dialogue between a human and a computer, two or more humans, or even between two or more computational agents. Embracing natural language processing techniques, components typically include parsing modules utilising language models. The application domain is modelled to some extent, be it a statistical model or a semantic network. Together these and other components work to comprise the dialogue management system. There are many examples of successful dialogue systems, from flight information services (Fraser 1995a) to theatre ticket booking (Hulstijn et al. 1996) and virtual space navigation (Nugues et al. 1996).

One of the earliest goals of Artificial Intelligence was to successfully emulate a human in terms of conversational ability. Alan Turing asked the question "Can a machine think?". Although he answered in the affirmative, he went on to ask the following question: "If a computer could think, how could we tell?". Turing's suggestion was that if the responses from the computer were indistinguishable from that of a human, the computer could be said to be thinking. This become known as the Turing test (Turing 1950).

Eliza is one of the earliest examples of a system which attempts to engage a human in a conversation. It was written by Weizenbaum in the 1960's (Weizenbaum 1966) and originally took the role of a non-intrusive psychotherapist. Eliza used simple pattern-matching techniques to respond to certain keywords and set phrases from the user. For example, if the user types in a sentence which includes the word "mother", then Eliza will respond with a question like, "Tell me about your mother". Eliza did not try to build a model of the dialogue, nor had any notions of what a dialogue actually is. This simple form of trigger-reaction to a sentence is obviously limited and it is unlikely that it would have passed the Turing test, despite several urban legends telling of people fooled by it.

Enthusiasm for the Turing test carries on, however, with the Löbner Prize (Löbner 1996, Shieber 1994). Began in 1990, in conjunction with the Cambridge Center for Behavioral Studies, it awards an annual prize for the most "human"-like computer program. The ultimate prize is for the first computer program whose responses are indistinguishable from a human's. Needless to say, nobody has won the grand prize, yet.

Dialogue management systems have also been around for many years in the guise of database query systems. A human can ask a computer either in natural language or in an unambiguous query language, such as SQL, for information from a database. Such simple Question & Answer systems have been in evidence since computers became widespread in businesses. Only recently, however, have systems which allow more sophisticated dialogues been available. The problem with natural language is that it can be ambiguous and once dialogue becomes more than a single question followed by a system response, discourse phenomena further complicate the interaction. The further the designer goes to make the dialogue appear natural, the more sophisticated the interaction. Ignoring discourse phenomena can lead to unnatural and unpleasant dialogues. Dialogue management systems are needed when user's requirements are spread over more than one sentence. They can also aid the identification and subsequent recovery from speech recognition errors or other misunderstandings.

By looking at the way we speak to humans, we can go some way to model this between a human and a computer. The research community is divided as to whether we should even try to model human characteristics on a machine. As various researchers, including Dahlbäck (1995), have pointed out, trying to build a system which is not differentiable from a human may be misguided. Humans show incredible resilience and ability to adjust to a dialogue partner. This is evident from our ability to adjust our speech to others with special needs, from children to non-native speakers of our language. Many argue that the effort of perfect simulation suffers from diminishing returns, since appearing human is very difficult and not actually necessary.

The remainder of this report is concerned with the features of discourse and dialogue management and the variety of techniques developed to model them. Note that the two terms "discourse" and "dialogue" are often used interchangeably by researchers, and are used interchangeably here.

2. The role of dialogue management systems

A dialogue management system can be used to simulate the process of a dialogue. Dialogue modelling is necessary for any type of dialogue, be it text-based, speech input or using other modalities. In the age of user-friendly interfaces, pleasant and easy interaction is an essential aspect of the design of any system. Dialogue systems have specific requirements for this, including adequate recovery from error. It is important that although your system may fail, it doesn't mean that the dialogue should, too. The dialogue manager should be able to identify errors and adopt a strategy which recovers the dialogue.

Dialogue management techniques are beneficial to systems using speech recognition. Spoken language dialogues require complicated modelling strategies, but these in turn can provide a level of constraint that can assist the shortcomings of speech recognition technology. A sentence can be anticipated to a certain extent by the discourse context. Such high-level, pragmatic constraint of the search space can significantly improve speech recognition performance (Churcher et al. 1996). Dialogue management itself can recover from any remaining ambiguity or recognition errors. This is analogous to our own experience of dialogue. Misunderstandings occur when one person mishears what is said, perhaps what is understood is not clear or is contradictory. Rather than abandoning the conversation at that point, both partners work together to clarify the misunderstanding. This form of recovery from errors is ideally suited to speech recognition, a technology which suffers from many 'misunderstandings' and is far from mature.

3. Features of discourse

Dialogue between humans shows a number of characteristics and whilst humans using a computer system may compensate by simplifying their own language, they still expect many of these characteristics to be retained. Some of these characteristics are:

- **overall structure**: looking at the generic structure, a dialogue has an opening, body and closing. Whilst the user may have control of the dialogue at its beginning and end, the system typically takes control in the body of the dialogue (see section below on initiative).

- **mixed initiative**: whilst the user may have the initiative for the majority of the dialogue, the system must be able to take control in order to confirm given information, clarify the situation, or constrain the user's responses if misunderstandings arise.

- **over-informativeness**: in dialogues between humans, one participant may offer more information than is actually required. Conversely, the system may offer information that hasn't been specifically asked for if it is considered helpful.

- **contextual interpretation**: users' sentences have to interpreted according to the dialogue context in order to correctly understand sentences which include ellipsis, deixis etc. (see section below on difficulties of modelling dialogue).

- **error recovery**: as speech recognition errors often occur and other ‘misunderstandings’ arise, the system must adopt a recovery strategy which allows the conversation to continue (see section below on error recovery).

For a detailed discussion of each of these characteristics and approaches taken to model them, see McGlashan (1996). The remainder of this section examines the phenomena in discourse which make modelling dialogue difficult, the structure of dialogue, initiative, the differences between spoken and written dialogue and a taxonomy for classifying differing dialogues.

3.1 Difficulties of modelling dialogue

Discourse includes a number of phenomena which need to be taken into consideration when modelling dialogue:

- turn-taking
- conversational fillers
- ellipses
- indirectness
- adjacency pairs and insertion sequences
- anaphoric references

Turn-taking

When examining turn-taking, speakers appear to alternate between speaking: agent A talks, stops, agent B talks, stops, A talks, stops and so on. In simple Question & Answer turns, agent A will ask a question which is answered by agent B. It is obvious when it is agent B’s turn to speak. Looking at less structured dialogue, it is less obvious when one agent’s turn is finished. Ervin-Tripp (1979) measured two agents speaking and found that considerably less than 5% of speech overlapped (an agent starts speaking before the other has finished). The gap between one agent stopping and another starting to speak often could be measured in a few tenths of a second. Turn-taking appears fluid and yet it is not obvious how speakers manage to determine when it is exactly their turn. This problem is further complicated by dialogues where more that two speakers are present. There has been some research which attempts to explain this phenomenon in terms of a set of rules which economically manage a shared resource (see Sacks et al. 1974, Schegloff & Sacks 1974).

Conversational Fillers

In conversation we often use what are known as *fillers*. These short phrases, such as ‘a-ha’, and ‘yes’, do not fulfil an act in themselves and yet are commonplace in conversation. They appear to add cohesion to a dialogue and re-assure the speaker that his or her partner is paying attention, following what’s said etc. Again, it is not obvious what prompts speakers to use fillers although they perform some function and contribute to a coherent conversation.

Ellipsis

Ellipsis commonly occurs in a sentence where for reasons of economy, style or emphasis, part of the structure is omitted. The missing structure can be recovered from the context of the dialogue and normally the previous sentences. This can be demonstrated in the following dialogue:

A: What time is Twelfth Night playing tonight?
B: It starts at 8:10pm tonight.
A: And Hamlet?

Without taking the context of the dialogue into account, it would be impossible to interpret agent A's second question. The actual derived question is 'And what time is Hamlet playing tonight?' and depends on the recent, in this case previous, sentence of A. Keeping track of the context is essential to coherent dialogue. Without modelling ellipsis, dialogue can appear far from natural.

Indirectness

Participants can often make sense of a dialogue even when there is little linguistic information present in the utterances. The literal meaning of a discourse is often not the same as the interpreted one. Participants use a wide range of cognitive skills interpreting an utterance and often draw on background knowledge of the discourse and the real world about them. Consider the following example first presented by Widdowson (1978):

A: That's the telephone.
B: I'm in the bath.
A: OK.

Taking a literal meaning of the utterances, the dialogue is not understandable, i.e. not coherent. However, if the discourse is interpreted given the range of real world knowledge of the participants, then it could be reformulated as follows:

A requests B to perform action (answer the telephone)
B states reason why he cannot comply with request (in bath)
A undertakes to perform action (answer telephone)

Such an interpretation of discourse as a 'social interaction' has led to varying non-literal treatments of discourse. One treatment is Speech Acts which are covered in section 6.1.1. This section also outlines further problems of indirectness or extra-linguistic context in discourse.

Adjacency pairs and insertion sequences

One of the first structures beyond the sentence level discovered by conversational analysts was *adjacency pairs*. Typical examples of these pairs are question-answer, greeting-greeting, offer-acceptance. Schegloff & Sacks (1973) provide a characterisation of pairs:

Adjacency pairs are sequences of two utterances that are:
(i) adjacent
(ii) produced by different speakers
(iii) ordered as a first part and a second part
(iv) typed, so that a particular first part requires a particular second

Adjacency pairs are intuitive structures which explain the structure of conversation. However, they do not adequately model *insertion sequences*, where adjacency pairs can be embedded within another pair. To illustrate consider the following example where a question is followed by a question:

A: Where are you going? Q_1
B: Why do you want to know? Q_2
A: I thought I'd come with you. A_2
B: I'm going to the supermarket. A_1

Agent A's first question, Q_1 is not answered, instead agent B responds with a question, Q_2. Agent A answers this, A_2 and then agent B answers the original question with A_1.

Other forms of insertion sequences have been found. Jefferson (1972) classified a number of different embedded sequences which do not follow the conventional adjacency pairs structure. For example, the *misapprehension* sequence occurs when the conversation is temporarily halted in order to clarify a point. In this case, the insertion sequence would consist of the misapprehension part followed by a clarification, followed by a termination sequence.

Unfortunately, there is no closed set of insertion sequence types. To further complicate matters, many are defined in such a way that makes them “transparent” - the linguist must subjectively determine which type applies for any dialogue, a process not assisted by the overlapping definitions. See Coulthard & Brazil (1992) for a full criticism.

Anaphoric reference

Another phenomenon which frequently occurs in dialogue is that of *deixis*. This is the linguistic term for words which can only be interpreted in the given context, such as ‘now/then’, ‘here/there’, ‘I/you’, ‘this/that’ etc. Expressions relative to the current context often need to be interpreted into an absolute or canonical form. For example, ‘next Tuesday’ will be represented as an absolute date depending on the date that the reference was made.

Words which refer backwards to previous concepts are known as anaphora, those which refer forwards are cataphora. Anaphora are common in dialogue for similar reasons as ellipsis: they reduce redundancy. Words such as ‘it’, ‘he’, ‘there’ allow speakers to economise their speech. Dialogue also involves non-linguistic language: anaphora can be augmented by gestures, or in the case of computer systems by other modalities, such as mouse clicks. To this end, sentences have to be interpreted according to their context, which must be modelled to such an extent that referring expressions such as anaphora can be fully resolved.

3.2 Structure in Discourse

There are two main approaches to defining the structure of a dialogue, as distinguished by researchers interested in either discourse analysis or conversational analysis. Conversational analysis, mentioned above, is concerned with reoccurring patterns in empirical studies of discourse. To this respect, this approach does not look at the structure of discourse as a whole, but rather its constituent parts.

There are a variety of ways to analyse the structure of discourse. The placement of discourse and pragmatics between semantics and sociolinguistics has provided a wealth of approaches, some taking a philosophical perspective, others a more traditionally structural approach. The super-sentence approach to discourse analysis is based upon the use of techniques employed by syntax and semantic grammarians. This approach tries to construct sets of well-formed sequences of utterances in discourse, presenting a dichotomy between ‘grammatical’, allowed utterances and ungrammatical utterances. To illustrate, consider the syntax grammarian who wishes to define a set of well-formed sentences. He/she may represent the set by a collection of context-free rules, which may start as follows:

S → NP VP NP (AdvP)

i.e. a sentence can rewrite as a Noun Phrase (NP) followed by a Verb Phrase (VP) followed by another Noun Phrase followed by an optional Adverbial Phrase.

In a similar fashion, a discourse analyst may construct a ‘discourse grammar’ which would define a grammatical discourse. Sinclair and Coulthard (1975) were the first to show that a discourse grammar was feasible. They started by recording a number of classroom interactions between a teacher and the pupils and went on to identify repeatable patterns in the discourse. They noted that a lesson could be broken down into a series of *transactions* which represented the current topic focus being discussed. Transactions could be broken down into *exchanges* between a pupil and teacher. Exchanges, in turn, are divided into *moves* which represent the components of the exchange, such as an *initiation* made by the teacher or a *response* by a pupil. Moves were then divided into *speech acts*, representing the function of the utterance. For further discussion on speech acts, see section 6.1.1.

Sinclair and Coulthard's 'system of analysis' defines a comprehensive internal structure to each level of the analysis for dialogue.

This approach has received criticism from many researchers in pragmatics whose main argument is the applicability of these techniques to other forms of discourse (for example, Leech 1983). Nunan (1993) demonstrates that these techniques can only be applied to some well-structured discourses, such as telephone elicitation tasks. Freeman (1992) found that even re-applying this analysis to his own classroom discourse resulted in greatly differing patterns. Applied to more general conversation, this approach runs into difficulties highlighted by the problem that there is seldom a one-to-one mapping between linguistic form and function in discourse. Nunan goes on to state that providing a grammar to restrict what is allowable beyond the level of a single utterance is 'almost impossible'. It is difficult to specify which utterances are and are not allowed at a particular point in the discourse.

Despite these criticisms, super-sentence approaches have been adapted and proved useful for dialogue management systems which tend to model highly structured tasks. As Dahlbäck (1995), Grosz (1977), Novick & Hansen (1995) all agree, the structure of dialogue is strongly related to the structure of the task being undertaken. Typical top-down constraints can be applied according to either the different 'phases' of a dialogue (Alexandersson 1996, Alexandersson & Reithinger 1995) or where an agent is following a goal-based approach, where sequences of sub-goals can be deduced. Rather than taking a rule based approach to structure definition, many researchers are using statistical modelling techniques to provide a level of flexibility (Alexandersson & Reithinger 1995). For a further discussion on the differing techniques used to model dialogue structure, see section 4.

3.3 *Differences between spoken and written language*

There is a distinction between language that is written and that which is spoken. Apart from a number of linguistic differences which are covered below, there is a difference in the type and structure of the discourse. The most prominent is in the their use of context. Context has two attributes: linguistic context and non-linguistic, or experiential context. Linguistic context is the language that comprises the discourse which is under analysis. Experiential contexts include such things as the type of communicative event, the topic, setting, time of day etc.

Spoken language tends to have a common situation to all participants, who rely on their background knowledge of the situation to interpret the discourse. Spoken language can also make use of paralinguistic clues such as eye contact, intonation and so on. Where one participant is a computer, similar clues must be modelled, as other modalities (for example, a mouse) may be used simultaneously. Written language can make limited use of a common situation and environment and hence must include enough information for the reader to infer these from the text. In some cases, however, the writer will make use of the reader's background knowledge.

The linguistic structure of spoken language is also different to that of written. In terms of differences in grammar, written language usually consists of clauses which are internally complex, making use of elaborate modifiers and structures such as dependent clauses. There is complexity in spoken language but this is normally present between clauses rather than within them. Naturally, this depends on the nature of the discourse; an informal email to a friend would be akin to spoken language, whereas a formal email to a conference organiser would show the characteristics of written.

Another distinction between written and spoken language is the number of content words (nouns and verbs) per clause. This is known as *lexical density* of the text, as defined by Nunan (1993). Written language usually has a higher lexical density.

3.3.1 Spontaneous spoken language

There have been a number of studies of spontaneous speech used in dialogues (Connolly et al. 1995, Fraser 1995a, Eckert 1996). Spontaneous speech differs from read speech in that the speaker is often greatly influenced by the environment. Not only will a speaker adjust to his or her dialogue partner, but he or she may use their words and even sentence intonation. Dybkjær et al. (1995) noted that whilst collecting dialogues in a WoZ experiment, speakers tended to use the words which appeared in the description of the scenario. To avoid this problem, Dybkjaer designed two sets of experiments: one used a text-based description of the scenario and task, the other a graphical representation of the same tasks. The graphic-based experiments negated the priming effect, but paradoxically generated a smaller vocabulary than the text-based approach. In a similar way, system questions and responses can influence the style of language of the user. Schillo (1996) also tried to avoid lexical "priming" by using graphical icons to prompt user responses. Schillo found a great deal of planning was required to "engineer" an icon-based dialogue and he had to resort to language for anything beyond basic functions.

A large proportion of spontaneous speech is ungrammatical, and yet it is astoundingly resilient to any arising errors (Connolly et al. 1995, Fraser 1995a). Any misunderstandings which arise, termed *communication deviations* by Taleb (1996), are resolved within 6 turns on average. Fraser notes that communication is inherently ‘messy’, and even goes as far as saying that messy communication may be a natural phenomenon and perhaps necessary for a pleasant, natural dialogue.

3.4 Dialogue initiative

At any stage in a dialogue, one participant has the initiative of the conversation. For example, when agent A asks agent B a question, agent A has the initiative. When agent B answers, however, agent A still has the initiative, i.e. has effective control of the conversation. Now, if agent B then asks a question of A, B takes control of the dialogue or takes the initiative. In a sense, the initiative is a function of the roles of the dialogue participants. Dialogue systems may allow the user or the system of take the initiative, or may allow both to switch roles as required.

Menu navigation, system-orientated question and answer systems where the system has the initiative throughout the dialogue are the simplest to model. In this case, the user must select between a small number of well-defined options. Where the user takes the initiative in systems such as command and control and database query systems more complex modelling strategies are needed. Although the range of tasks to be performed may be low, the user has greater expressibility and freedom of choice. The hardest dialogues to model, by far, are those where the initiative can be taken by either the system or the user at various points in the dialogue. As noted by Eckert (1996), mixed initiative systems involve dialogues which approach the intricacies of conversational turn-taking, utilising strategies which determine when, for example, the system can take the initiative away from the user. For systems which use speech recognition, the ability to confirm or clarify given information is essential, hence system-orientated or mixed initiative must exist.

3.5 A dialogue taxonomy

Dahlbäck (1995) argues for a classification or taxonomy of dialogues. Comparisons of different dialogue management techniques or computational theories of discourse are difficult when dialogues have differing features. If dialogue systems could be classified according to some taxonomy, then any algorithms developed could be applied to different systems with comparable results. Dahlbäck's taxonomy is based on work by Rubin (1980) and Clark (1985). It distinguishes four main dimensions:

1. Type of agent
2. Channel
3. Task type
4. Shared knowledge

The type of agent, either human or computer greatly influences the style of spoken language used. Work by Guindon (1988a) and Kennedy et al. (1988) shows that when addressing a computer, human utterances are shorter and have a smaller lexical variation with a minimal number of pronouns.

The channel of communication involves many different aspects. The most obvious is the modality of the channel which may be either written or spoken; the distinctions between the two have been explored above. Other aspects of the channel include the style of interaction (influencing for example, the anaphor-antecedent relations), spatial and temporal commonality, concreteness of referents (are the objects and events referred to visually present or not) and separability of characters, all of which influence the dialogue structure and style. For example, when Guindon (1988a) analysed dialogues of users using a statistical computer package, occurring pronouns referred either to their antecedent in the current sub-dialogue, or to the package which was on the screen.

As noted above, the dialogue structure is influenced by the structure of the task (Grosz 1977). Grosz points out that some dialogues are less influenced by their task structure than others. This is not only dependent on the task itself, but also the task setting. The number of different tasks managed also affects the dialogue structure. Unsurprisingly, Dahlbäck found that a more complex topic management was required when the user could undertake multiple tasks in the same dialogue, such as information retrieval and reservation booking (Jönsson 1993, Ahrenberg et al. forthcoming).

Clark (1985) showed that different kinds of shared knowledge between dialogue participants influences language in dialogue. Clark asserts that participants use a number of heuristics to obtain an understanding of the dialogue through mutual knowledge. There are three types of information which can be used to infer the common ground between speaker and listener: shared perceptual, linguistic and cultural knowledge. The first two types are the easiest to model. When both participants are aware of physical or visual objects then they share perceptual knowledge. Linguistic knowledge is the shared knowledge of the dialogue up to that point. Cultural knowledge is more difficult to model since it depends on knowledge that is obtained from the participant's community.

4. Approaches to dialogue management

A dialogue manager must model to a certain extent two aspects of the interaction between the user and the machine. The first of these is the interaction history which allows user's sentences to be interpreted according to the current context. The context may also deal with phenomena such as anaphora and ellipsis. The second aspect is a model of the interaction, controlling the strategy adopted by the system to control the dialogue structure. This strategy also influences the role of initiative in the dialogue.

There are many approaches to the design of dialogue management systems. Typically, a system will consist of the components in Figure 1, below. Text or transcribed speech is input into the system which is parsed and semantic information deduced by the natural language components. The dialogue manager then co-ordinates between the other components in order to fully understand all the implications of the input in the current context, resolving conflicts, ambiguities and dealing with anaphora and ellipsis. Multi-modal systems can receive from and output to many different devices. The response generation module generates a response based on the output of the dialogue manager, usually as a natural language text to be read out by the speech synthesiser, but may also generate graphical or other output. Whilst the system may output text to be converted into speech, it may also include visual output. Each device needs a level of abstraction, a driver, which interprets between the actual device and the dialogue manager, so that all input and output can be modelled with the same mechanisms.

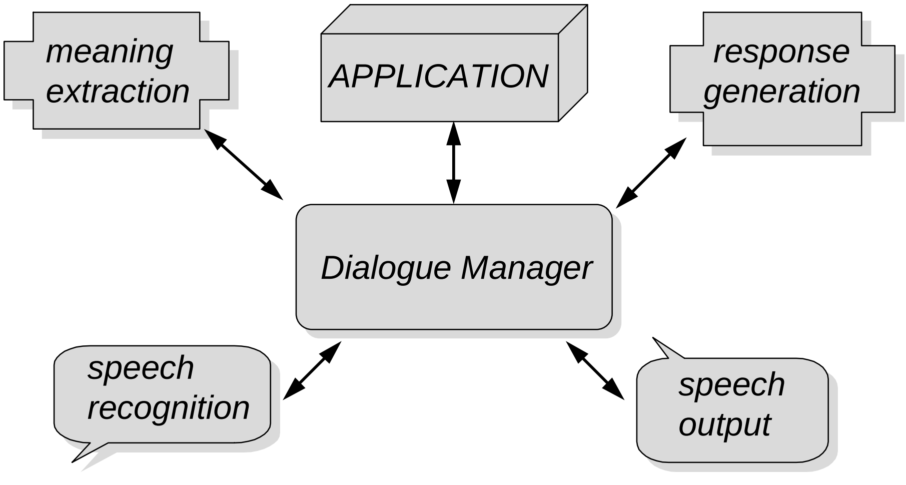

Figure 1: Overview of a generic spoken language system

Figure 1 is a basic overview of a generic spoken language system. Figure 2 below, expands on this and provides a broad outline of a detailed, generic model, reflecting the key components required for a number of functions including discourse phenomena such as anaphoric and ellipsis resolution and a model of the task structure and how it relates to the structure of the dialogue. As is the predicament of any generic system, it is necessarily vague and since it attempts to combine components found in a variety of individual models, it may not fit all systems, if any in particular.

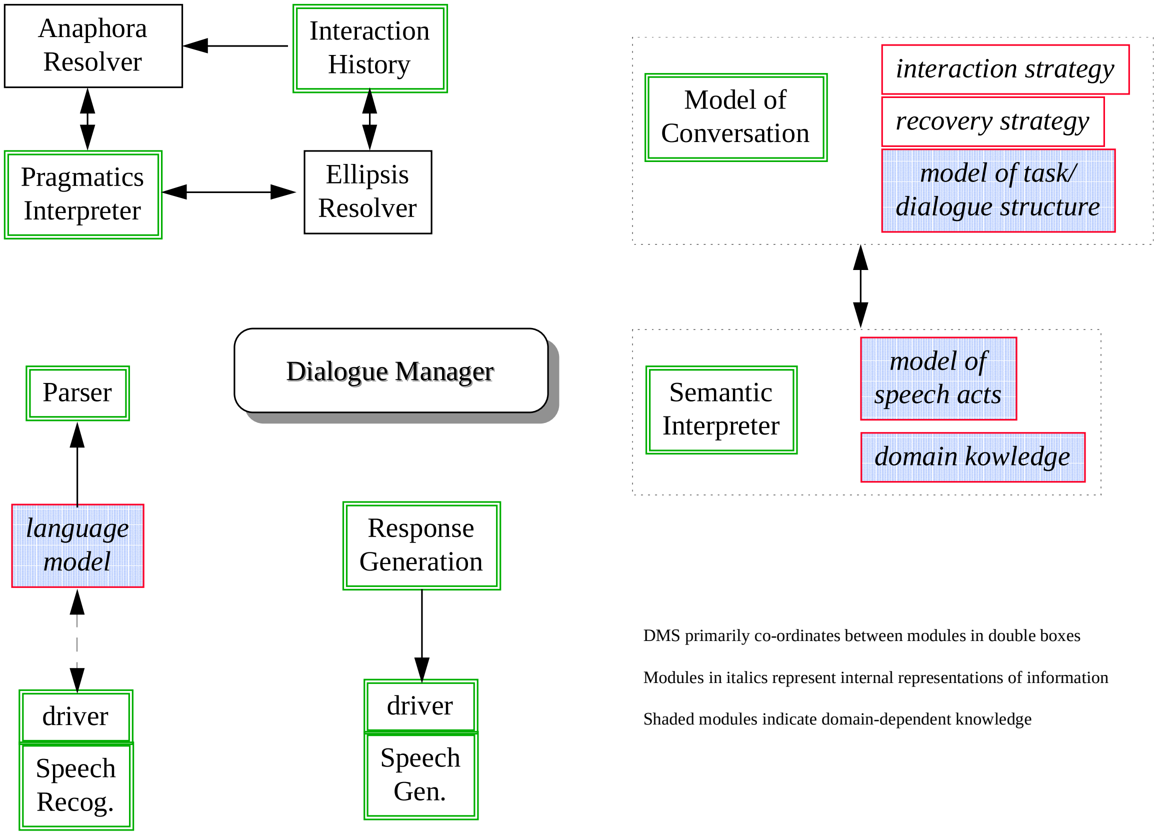

Figure 2: Outline of key components in a spoken language dialogue manager

The design of dialogue management systems involves the design of each of the components within the system. Whilst some researchers have concentrated on general approaches to dialogue management, others have developed specific strategies for individual components. Three main approaches are described here: dialogue grammars, plan-based approaches and collaborative dialogue. They are certainly not mutually exclusive approaches and are often used in association with the others.

This and the following sections discuss a combination of modelling strategies for discourse and their implementation issues. For a summary and discussion of the three main approaches to discourse modelling: semantics (not covered here), structural and intentional, see McKevitt et al. (1992).

4.1 Dialogue Grammars

One of the first approaches to the representation of dialogue was the use of a prescriptive grammar for sequences of sentences in a dialogue. Often the grammar would describe the structure of the complete dialogue from beginning to end, although recent approaches have taken a less super-sentential approach by describing commonly occurring sequences in dialogues, for example, *adjacency pairs*. By creating a type of grammar, whether a graph or finite state machine or a collection of declarative rules, a form of top-down structure can be imposed.

Levinson (1981) proposes a number of arguments against such an approach to a general theory of dialogue. Notably, a sentence can perform more than one communicative function and a speaker may expect a response to address more than just the most recent context. Levinson argues that the system would have to model multiple simultaneous states, which is not feasible for a finite state machine. Conversational Games Theory, described below, goes some way towards this problem by allowing more than one communicative act to address more than the immediate context. Further extensions intend to allow multiple speech acts from one sentence to influence the context. Levinson also argues that indirect communicative acts, where what is meant is not the literal interpretation of what has been said, are impossible to model without some form of calculus for these acts. Maier (1996) provides an interesting approach by statistically modelling sequences of communicative acts in order to derive the indirect intention behind them. This does not, however, adequately model the phenomenon of indirectness arising purely from the extra-linguistic context, i.e. the situation. The popular example of "It's cold in here" being interpreted as a request to 'close the window', can best be modelled by plan-based approaches.

Graphs and finite state models

Graphs and finite state models are the simplest of dialogue managers. They are ideally suited to applications where the dialogue structure is the same as the task structure. Graphs consist of a series of linked nodes, at each of which is a system prompt detailing a limited number of choices the user can make at that point. Each allowed response leads to another node. Graphs offer a number of advantages over other approaches. As the system takes the initiative all of the time and there are only a few choices presented to the user, then it is a simple matter to anticipate the user's response, a technique useful to speech recognition further helped by the priming effect of the system prompts.

Although there have been examples where graphs and finite state machines, on their own, have been successfully used (Muller & Runger 1993, Nielsen & Baekgaard 1992), they have received criticism over their lack of flexibility both to deviations in the dialogue structure, and also in their applicability to other domains (e.g. Aust & Oerder 1995).

To compensate for the graph's lack of flexibility, a number of systems have taken frame-based approaches, which are based on the structure of the entities in the application domain, often called the thematic structure or topic. By modelling a hierarchical set of entities the system can control the dialogue according to the requirements of those entities. For example, in Hulstijn et al. (1996), a theatre booking system, the details of the performance the user wishes to see can be thought of as the topic of the conversation. The topic has a number of slots such as the date of the performance, title, time etc. Furthermore, certain slots can become topics unto themselves with a set of slots and attributes. In Hulstijn's system the frames are arranged hierarchically to reflect the dependence of certain topics on others. Veldhuijzen van Zanten (1996) demonstrates a similar system where a frame structure relates entities in the domain to one another in a train timetable information system. From the topmost entity of the user to individual details of travel times and places, the structure captures the meaning of all possible queries the user may make. The dialogue manager uses this information to construct database queries. A set of rules indicate what information is required before the database can be queried which in turn drives the dialogue manager to elicit further information from the user.

Similarly, Aust & Oerder (1995) uses a set of rules to define well-formed database queries and uses this to drive the dialogue. The dialogue component of the Philips Automatic Train Timetable Information System uses a declarative approach to model the dialogue structure, rather than rely on the structure of frames. By defining the information needed for a database query, the dialogue component can cycle through a simple process of checking for ambiguities, filling in slots for a database query and maintaining discourse continuity. The declarative method chosen lends itself well to other domains, since the structure of the dialogue is controlled by a separate declarative definition of valid database queries and system directed prompts to elicit the appropriate information from the user. Aust makes a distinction between the dialogue process and its declarative description, represented in a Dialogue Description Language (DDL). Baekgaard has had similar ideas for a DDL, aiming to create a generic dialogue system which refers to a domain-specific description of the dialogue. The system has been adapted for a number of applications including a flight ticket reservation system (Baekgaard et al. 1995, Bernsen et al. 1995, Dalsgaard & Baekgaard 1994) and a telephone based automatic banking services (Larsen 1996). As Grosz (1977) points out, there is a continuum of the similarity between the task and the dialogue structure. Whilst some designers distinguish between the task and the dialogue, others embrace the two together. For the systems described by Baekgaard, the structure of the task is close to that of the dialogue. In these cases the DDL models the dialogue and task structure whilst the generic dialogue manager navigates through this description.

The system uses three levels of abstraction to describe the dialogue structure. The first is a textual level which interprets incoming events such as a spoken sentence or a mouse event. This information is translated into a set of object-value pairs for the next level, the frame level. The final level is a graphical description of the dialogue using a set of flow-chart style symbols standardised by CCITT (Belina & Hogrefe 1988). The dialogue flows according to conditions attached to the symbols and the object-value pairs obtained from the frame level.

Whilst some researchers advocate a generic dialogue system which is separate from the structure of the dialogue itself, others argue against abstraction claiming that since it is so closely related to the task structure, it is difficult if not impossible to generalise. Novick & Sutton (1996) takes the opposite approach to abstraction by building libraries of common dialogues, such as asking for the time, date and so on. The authors believe that libraries are better for the developer since they can encompass relevant dialogue management expertise. As they point out, the developer may be an expert in his/her application domain, but they may not be an expert in dialogue management techniques. System prompts and responses need to be carefully tailored so that the user responds in a controlled and anticipated manner (see section 5). The libraries can be customised for the application and combined with other sub-dialogues to describe the structure of the overall dialogue. The libraries allow the reuse of code and expertise of researchers in dialogue management. One criticism though, is that analogous to the rapid prototyping of Graphical User Interfaces, designers can still create poorly designed systems, despite a wealth of customisable tools.

4.2 Plan-based approaches

Plan-based approaches show a greater complexity to dialogue modelling than discourse grammars. Taking the view that humans communicate to achieve various goals, plan-based approaches attempt to model these goals, and the sub-goals which comprise them. It is the task of the listener to discover and appropriately respond to the speaker's underlying plan. Taking the example from Cohen (1996), in response to a customer's question of "Where are the steaks you advertised?", a butcher's reply of "How many do you want?" is appropriate because the butcher understands that the customer's underlying plan is to buy certain steaks. By being co-operative the butcher adapts to the plan of the customer so that the plan may be successfully achieved. Plan-based systems can adequately deal with indirect speech acts as the example shows, on the precondition that the plan has been correctly identified.

The Verbmobil project uses a number of approaches to modelling dialogue, including finite state models of sequences of speech acts and plan-based operators (Alexandersson & Reithinger 1995, Maier 1996, Alexandersson 1996). The system represents the intentional structure of the dialogue on four levels: the lowest dialogue act level representing the speech acts of the speaker; a turns level which models more than one speech act within an actual turn; a dialogue phase level indicating the phase of the dialogue (e.g. opening, negotiation, closing) and the dialogue level at which individual dialogues take place. The dialogue is parsed from overall goals to sub-goals by a set of plan operators derived from an annotated corpus of example dialogues. The plan operators are derived using three steps:

1. Simplification of corpus: dialogue acts are made domain independent, multiple sequences of acts are collapsed into one occurrence.
2. Sequences of dialogue acts are generalised into a small set of classes.
3. A plan operator is generated for each class of turn using the Bogus System (Stolcke 1994a, 1994b). This generates a context-free stochastic grammar using bayesian model merging.

Initial predictions of a subset of the original corpus show that plan operators can be used with relative success, although there are a number of problems inherent in plan-based approaches. The first is the difficulty of predicting with accuracy which plan is currently in use by a user. This is related to the inherent ambiguity and lack of definition of the mapping between linguistic form and function, as discussed in section 3.2. Again this is dependent on the functionality of the application. Going back to the example above of the customer and the butcher, the butcher interprets the intention of the customer by identifying the relevant plan. The actual interpretation is quite different from the illocutionary act and it is not clear how the butcher identified the plan from it.

Plan-based approaches lack a sound theoretical basis and do not capture why participants are motivated to speak in a dialogue nor for the occurrence of phenomena such as clarification sequences. Plans rely on the task structure but need to incorporate some form of meta-level in order to show how plans can be manipulated when clarification sequences and so forth arise.

4.2.1 Conversational Games Theory

Conversational Games Theory (Carletta et al. 1995, Kowtko et al. 1991) is an interesting scheme for the coding of dialogue structure in a corpus, which can also be successfully used to model the conversation between a human and a computer in a task-orientated dialogue (Williams 1996a). It uses techniques from both discourse grammars and plan-based approaches: its structural approach includes a goal or plan-orientated level. Each game consists of a set of partially anticipated turns, such as a response following a question. The scheme is comparable to that proposed by Sinclair and Coulthard (1975) to model the structure of classroom lessons (see section 3.2):

Sinclair & Coulthard	**Kowtko**	
Lesson	Dialogue	
Transaction	Transaction	Related to the current sub-task/goal
Exchanges	Games	Complete interactional unit: e.g. Question and answer sequence
Moves	Moves	Comprise speech acts, e.g. QUERY-YN, including task-orientated acts and those responsible for dialogue control.

Since the scheme was applied to task-orientated dialogues, each dialogue consists of a number of small tasks which work together to complete the overall task. A dialogue then, comprises one or more transactions, each of which represents a sub-task. Each transaction comprises a series of conversational games. A game consists of an opening move and in some cases optional, end move. For example, a QUERY-YN game would start with a QUERY-YN move followed by either a REPLY-Y or a REPLY-N move. To account for discourse phenomena such as side sequences, clarifications and so forth, each game can have another game embedded within it. Such a recursive structure can model the complexities of natural dialogue to a high degree.

Originally developed as a coding scheme for the HCRC Map Task Corpus (Anderson et al. 1991) for the purpose of statistically comparing intonation contour with the conversational move function, it has since been adapted to a number of spoken language dialogue systems (Lewin et al. 1994, Bird et al. 1995). Although the coding scheme was applied to corpora subjectively, Kowtko showed that novices could re-apply the scheme with accuracy between 78% and 88%. Also, the move classes could be successfully replicated for other domains (Carletta et al. in prep).

Focusing on a particular implementation of the theory, BT’s MailSec (Williams 1996a) uses a dialogue manager to co-ordinate a conversation. MailSec does not implement the transaction level of the scheme, perhaps because of the low task complexity of the system. MailSec is designed for the user who is out of the office but still wants access to email. The system allows the user to call a computer and engage in a dialogue to list or read out their emails, or other typical tasks.

The dialogue manager takes in a transcription of the user’s sentence from a continuous speech recogniser and then co-ordinates with other modules to build a database query. The response from the database is interpreted and passed onto a text-to-speech module for the user to hear. Of particular interest is the conversational model consisting of a history list of previous moves and a games stack, necessary to model the recursive nature of the dialogue. The history list is used for reference and ellipsis resolution by searching for the most recent entity which matches the necessary requirements. For example, the occurrence of “his” in a sentence will cause the system to search the history list for the most recent entity referenced which is a human male.

The system has a number of interesting features. Firstly, when a database query returns nothing, then the system compensates by relaxing the constraints of the query. To illustrate, take the following example from Williams (1996a):

> Caller: Do I have any emails from Anna about elephants?
>
> MailSec: No, you have no emails from Anna Cordon about elephants, but you have one message from Keith Preston about elephants.

Here the system has found no emails in the database that are from Anna and are about elephants. It tries to be co-operative by arbitrarily relaxing one of the constraints, in this case that the emails have to be from Anna. The choice of which constraints to relax is a contentious issue and depends on the application.

The system achieves a certain degree of mixed-initiative by allowing users to respond to system questions as they see fit. It is possible to ignore a system question in order to take the initiative and direct the conversation as the user wishes.

Conversational Games Theory provides a method of modelling mixed-initiative, complex dialogue. By allowing one move to terminate multiple, nested games it answers some of Levinson’s (1981) criticisms of structural approaches, although it does not model sentences which contain more than one function. Williams' approach has ignored the transactional layer which represents the sequences of sub-tasks, effectively taking the discourse grammar approach to dialogue.

4.3 Collaborative approaches

In order to explain the process of dialogue in terms of the interactions of both partners, a number of ideas have been proposed for viewing dialogue as a collaborative process where both partners work together to achieve a mutual understanding of the dialogue. Regarding dialogue as a joint activity places certain commitments on both partners to understand one another. These commitments motivate discourse phenomena such as confirmation and clarification, evident in human to human conversations.

Rather than concentrate on the structure of the task as plan-based approaches tend to do, collaborative approaches try to capture the motivation behind a dialogue and attempt to capture the mechanisms of dialogue itself. Plan-based approaches do not explain why we have dialogues, or even why we do not simply walk away in the middle of one. Typically, the system will model both participants beliefs of the conversation, one agent will make a proposal which the other either accepts or rejects. Once accepted it becomes part of the shared beliefs of both agents. Typically the proposed goals of the agent will be adopted by the partner, together they work co-operatively to achieve those goals. This section shall continue by describing the main features of several such theories. Since each approach uses a combination of techniques from agency theory, plan-based and discourse grammar approaches, a brief overview of the collaborative aspects are described.

Novick & Hansen (1995) and Novick & Ward (1993) propose a model of collaboration for the establishment of mutual information, extending the theories of Clark & Schaefer (1989). Agents collaborate in building a mutual model of conversation and shared belief using a set of domain dependent and independent speech acts. The belief system, Figure 3, is similar to that proposed in Allen (1991), Traum (1991), Traum & Hinkelman (1992). Agent A draws on his/her own beliefs as well as what he/she believes agent B to believe in order to form an intention to act. This act is carried out in the form of an action, usually an utterance. The action is then interpreted as a specific act by agent B according to his/her personal beliefs and what he/she believes agent A to believe. Although both agent’s belief states appear separate from one another, Novick has proposed a level of mutual, i.e. shared knowledge.

This theory has been implemented for a number of domains. The agent’s beliefs are modelled using a set of propositions for each agent of the form, believe (*agent*, *proposition*, *truth-value*, *mutuality*). The agents can also have intentions to perform acts: act (*agent*, *proposition*, *truth-value*, *mutuality*). Control of the dialogue is by the execution of rules which deal with establishing information as mutual, expectations of an agent’s reaction to an act, domain-dependent planning rules and a rule that ensures intended acts are actually performed.

By establishing a number of initial beliefs and intentions, the system is able to simulate conversations which are comparable to those occurring in collected human to human dialogues. Two domains of collaborative tasks have been simulated: the conversation between the Air Traffic Controller and a pilot conducting an Instrument Landing System landing (Novick & Ward 1993); and a letter-sequence completion task where the letters are distributed among the two participants (Novick & Hensen 1995).

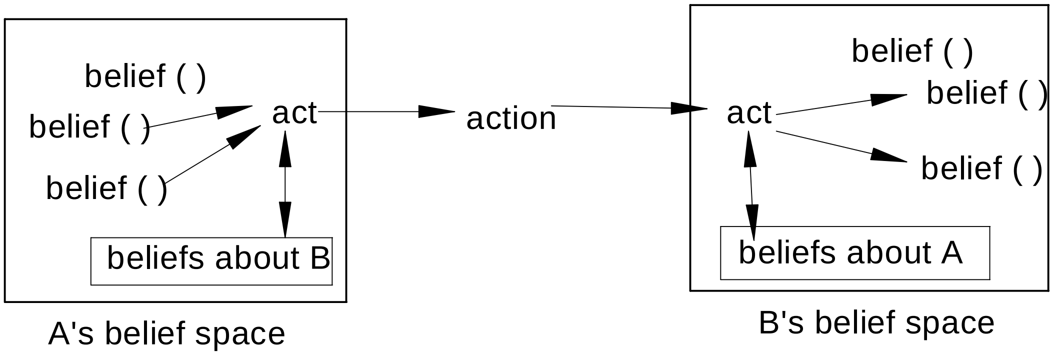

Figure 3: Agent's belief states

Traum's model of conversation agency (Traum 1996) extends Bratman's et al. (1988) Beliefs Desires Intentions (BDI) agent architecture by incorporating the concept of mutual belief, i.e. what both agents believe to be true. The BDI model is a popular formulation of agency, where actions in the world affect the agent's beliefs. The agent can reason about its beliefs in order to formulate desires and intentions. Whereas beliefs are how the agent perceives the world, desires are how the agent would like the world to be. Intentions are formulated plans which realise these desires. Traum criticises the BDI model and cites two main problems which he later uses to extend the model. Firstly, in the BDI model the agent's perceptions only influence its beliefs. Traum argues that an agent's perceptions also influence its desires and intentions. Secondly, the BDI model does not support more than one agent. By extending the model and introducing mutual belief, Traum has successfully implemented conversational agency in the TRAINS-93 dialogue manager.

Similar approaches to simulating collaborative dialogue have also been developed by Chu-Carroll (1996) who extends Sidners' work (Sidner 1992, 1994). Sidner's system allows an agent to make a proposal; if it is accepted by the partner, then it is integrated into the mutual belief space of both agents, otherwise the proposal is abandoned. Rather than rejecting the proposal outright, Chu-Carroll's work allows the agent who made the proposal to enter into a negotiation. The model recursively evaluates the agent's beliefs attempting to provide arguments to convince the other agent of the proposal. The model is different from other argumentative models (Allen 1991, Grosz & Kraus 1993, Grosz & Sidner 1990) in that it is not completely orientated to winning the argument. The model makes use of Gricean Maxim of Quantity (Grice 1975) where only enough information is presented to be convincing. A proposal consists of a belief tree where suppositions are supported by beliefs of varying strengths. By identifying the most significant beliefs the system can construct an argument which addresses the most important concepts. The argument is recursive and hence can be viewed as a negotiation.

Beun (1996) uses a set of generation and update rules applied to agent's beliefs to control dialogue flow. In this model one agent asks a question and both agents work co-operatively to answer this question. Conventionally, both agents would be able to perceive the world or application domain about them. Beun takes a simpler approach by ignoring the world: all of the information required to answer the question is distributed between the agents. Similarly to Novick above, each agent maintains what it personally believes to be true, what it believes the other agent wants to know and what it believes the other agent does not believe to be true. The generation rules which guide the discourse follow the Gricean Maxims of Co-operation (Grice 1975), where an agent cannot ask whether something is true if they already believe it to be true, nor if they know that their partner believes it to be false. These rules generate communicative acts which trigger update rules for both of the agents, in turn changing the belief states of each agent.

The rules allow complex dialogues to take place. Since the information is distributed, sub-dialogues can occur and counter-questions asked. After implementing the theories and running simulations of dialogues, the author found that although the sequence of communicative acts was typical, the resulting dialogue was not very natural and could be embellished with additional cohesive elements. The implementation combined communicative acts with a propositional content (e.g. SUGGEST_DATE act could contain the suggested date). This approach has an advantage over speech act grammars in that it automatically takes the context into account. The author acknowledges that the incorporation of a mutual belief space could reduce redundancy when referring to common objects, increasing the naturalness of the dialogue.

The use of principles to motivate dialogue is also used in the Constructive Dialogue Management (CDM) framework (Jokinen 1996). CDM views dialogue as negotiation between two agents who collaborate to establish *mutual knowledge* in order to fulfil an agent's goal. Dialogue as negotiation is in contrast to the one-shot Question & Answer style of dialogue (e.g. Moore 1995) even where the system's response was deemed to be co-operative by clearing up misconceptions and anticipating future responses. The framework determines the system's response from three interrelated factors: principles of ideal co-operation (Jokinen 1995), a joint purpose or communicative strategy and a set of communication obligations. Taken together they account for discourse phenomena such as vagueness of parameters leading to clarifications, side-sequences and so on.

The principle of ideal co-operation states that agents:

1. voluntarily strive to achieve the same purposes (i.e. system adopts user's goal only if relevant),
2. actions (communicative acts) do not hinder another agent's goals,
3. expect another agent to adhere to principles 1 and 2 *unless* explicitly stated.

Communicative obligations fall into three groups: sincerity, motivation and consideration. Sincerity ensures that agents tell the truth as they see it and do not try to deceive other agents. Motivational obligations allow the agent to justify whether something can be said and place obligations on the agent to offer information addressing the partner's goals. Consideration obligations are social constraints which enable the conversation to be as co-operative as possible, including:

1. if the partner's goals cannot be fulfilled then state why
2. if the quantity of information is too great then it should be split up
3. repetition of information should paraphrase original or use ellipsis
4. if information is irrelevant then tell partner so
5. if partner has initiative and unfulfilled goals then allow partner to change topics if desired
6. include related information even if not asked for when closing topic

4.4 Summary of approaches

Over the years there have been increasingly complex strategies for modelling dialogue. The breakthrough by Grosz that the dialogue structure is often dependent on the application domain, initially pushed people away from generic dialogue systems. Discourse grammars and plan-based approaches reflect this direction. Recently, collaborative, agent-based approaches have enabled dialogue to support more of the discourse phenomena noted above. Whilst acknowledging that the dialogue structure is still task-dependent, collaborative approaches attempt to capture the generic properties of dialogue. Despite successes in certain restricted domains, there has been limited success in others. The paradigm relies on modelling all of the participants' beliefs pertaining to the domain. Inferencing over these beliefs may occasionally fall outside the scope of the domain, leading to an incomplete or inconsistent model. Hence, the complexity of creating a collaborative system may lie in the definition of the scope of the domain. Understandably, the more complex and natural the dialogue is, the greater the chance that a speaker will use language in an unanticipated way. The success of plan-based approaches where the system holds the initiative lies partly in the ability to constrain the user's response. The trade-off is that the dialogue looses a certain amount of flexibility and naturalness. Hence, there is still a great deal of interest in mixed-initiative dialogue grammar and plan-based approaches.

5. Common approaches to building a DMS

In this section we will explore the common processes of building a dialogue management system. A popular approach to is to use iterative Wizard of Oz (WoZ) experiments to elicit the language and refine the interaction between system and user. This is typical for human/machine dialogues, where the system needs to control the dialogue to a certain extent. Wizard of Oz experiments are often used to determine the user's behaviour towards a system when the system is still being developed. Inspired by the film, The Wizard of Oz, a human imitates the computer and the user believes that they are interacting with a real system. This process can be a useful part of the design process.

Dybkjær's et al. (1995) approach is similar to others, including Fraser and Gilbert (1991), Fraser (1995a), Hulstijn et al. (1996). In this instance, Dybkjaer use a series of Wizard of Oz experiments to elicit the language of the domain, in this case a flight reservation system. The process begins by examining a collected corpus of human to human dialogues in the scope of the domain. A set of increasingly specific WoZ scenarios are created using a stand-in human who takes the place of the computer. Users are led to believe that they are communicating with an actual, automated system. By interviewing the users after each use of the system and measuring how difficult they found the interaction, the designers can build up a picture of how successful the system is. The iterative process continues until two objectives are reached: the users found the interaction comfortable and user's responses could be anticipated. By tailoring the system's questions so that the user's response is limited, language closure can be achieved. Dybkjaer reported that this took seven iterations to complete.

The flight enquiry system was entirely system-directed, that is the system took the initiative throughout the dialogue. The design of the WoZ scenarios was aided by the inherent structure of the task. Dybkjaer reapplied the elicitation methodology to a flight reservation system where the initiative is mixed but was forced to abandon the process because of the lack of structure of the task. Fraser (1995a), however, applies a similar process to a minimally mixed-initiative domain. The difficulty may lie in devising scenarios which encompass the entire domain. As the structure is relaxed there are many more combinations of utterances and hence an exponential increase in the number of possible scenarios. This is one argument against the use of WoZ experiments.

Vilnat (1996) raises another objection to the use of WoZ experiments. She questions the validity of the whole process by asking whether the user is sufficiently motivated to speak in an authentic manner. Whilst acknowledging that a collected corpus can be useful to the design process, designers should be aware of the limitations of such an approach. Providing a structured scenario can produce different language than from a user who is properly motivated to use the system. Additionally, the human wizard must know the limitations of a computer system: the problems detected by the wizard may not be the same as those experienced by the system. For instance, it is quite difficult for the wizard not to understand a request, or to misunderstand it in the same way that a computer might.

The design of scenarios raises an interesting point. Several researchers have noted that system responses influence the language of the user (see Eckert 1996, Dybkjaer & Dybkjaer 1993, MacDermid 1993, see also section 3.3.1). The method used to present a user with a scenario can similarly influence the user's language. Since the purpose of the experiment is to elicit the language that the user would actually use, the notion of priming invalidates the experiment. To compare and contrast the differing effects of priming, Dybkjaer developed two ways of describing a scenario. The first was entirely text and consists of a description of the environment followed by a description of the task. The second combined a textual description of the environment but conveyed the actual task using graphics which were designed to be non-priming. Interestingly, they found that the graphic-based scenarios produced a smaller vocabulary than those that were text-based. One explanation of this could be that the users would use their own choice of vocabulary for the graphical scenarios, but used a combination of their preferred vocabulary with the words primed by the text scenarios. Graphically based scenarios appear to provide a good method for eliciting the user's language rather than replicating the designer's (see also Schillo 1996). Yankelovich (1996) provides some interesting points on designing system prompts, some of which are discussed in section 6.3, below.

Corpora of dialogues are being increasingly used in the design stage. They provide a first basis for designing WoZ scenarios. Corpora have been used to directly derive a language model of the domain. The use of human to human dialogues to model those between human and computer is questionable, because of the different style of interaction. Such models have been successfully used in the Verbmobil project (Alexandersson & Reithinger

1995, Alexandersson 1996, Maier 1996) although in this system, the computer monitors human to human conversation and plays no active role. Section 6.1 discusses the possible coding schemes which can help elicit language structure and regularities.

6. Special Issues relating to a DMS

This section examines two areas of special interest of dialogue management systems: the coding of dialogue structure in corpora and beyond, and the recovery strategies employed for certain types of error occurring in a dialogue.

6.1 Coding Schemes

Coding schemes go beyond the structural annotation of corpora and are often used to model conversation either after a deterministic fashion, for example in a finite state machine, or non-deterministic fashion, as a statistical model. How much information can be encoded in a dialogue? The answer lies partly in the coding scheme and the style of modelling employed. Despite the many different schemes and uses for these schemes, most rely at the base level on the communicative function, i.e. what meaning the user wants to convey by an utterance. This mapping from the utterance to the intention can be a difficult process and is aided by the context. Work in this area was started by Austin (1962) and later developed by Searle (1969, 1985) into a theory of ‘speech acts’. Many coding schemes are based either on speech acts or a derived equivalent.

6.1.1 Speech Acts

Speech Acts were first brought to the fore by Austin (1962), a linguistic philosopher. Searle (1969) developed the ideas further. Austin had noted that some sentences can be considered as ‘performing an act’. They are not statements about the world but utterances which perform an action. He quotes two sentences as illustrations:

> “I bet you sixpence it will rain tomorrow”
> “I name this ship the Queen Elizabeth”

The two utterances cannot be reduced into terms of truth/falsehood in the same way as the utterance, “The sky is green” can. Austin classed these types of utterances *performatives*, which can be either ‘happy’ or ‘unhappy’ according to a set of *felicity conditions*. These conditions are based on whether the utterance can be effective as the speaker actually intended. Austin contrasts performatives with constatives which can be reduced to true/false values.

The dichotomy between performatives and constatives was later shown to be false. Both types of utterance can be classed as special cases of *illocutionary acts*. The illocutionary force being what the utterance actually does, for example to offer something, promise or warn somebody. The illocutionary act is termed a **speech act** and reflects what an utterance does from the point of view of the speaker.

A problem arises for speech acts when the derived act is not the actual intended act of the speaker. English frequently makes use of utterances which should not be taken literally. In this case the *perlocutionary act* must be identified. The popular example of “It’s cold in here” seen not as a statement but a request to close a window, turn up the heating or whatever, illustrates the problem well. Searle (1975) classified these as a special kind of speech act: an indirect speech act (ISA). ISA’s present problems for analysts assigning a speech act to an utterance since it relies on the interpretation of what the speaker intends. The ambiguity of whether an utterance should be taken literally or interpreted makes automatic labelling very difficult.

Brown & Yule (1983) argue that there are additional problems when assigning speech acts to utterances. A speech act is not a linguistic unit which fits directly into the conventional structure. Taking the speaker’s point of view, there may not be a one-to-one mapping between the utterance and an act. The speaker may use several utterances to perform just one speech act. Other times a single utterance can perform multiple speech acts. For example, consider the following utterance:

“Hey, Michele, you’ve passed the exam”
(Brown & Yule 1983)

If the speaker is Michele’s teacher, an obvious interpretation is that the speaker is making an assertion. If the speaker is Michele’s husband then the utterance may be an ‘asserting’ act, but could also be ‘congratulating’, ‘apologising’ (if he had had doubts) etc.

Despite these limitations, speech acts are used extensively in one form or another in dialogue management systems. Before being able to apply speech act labels to a dialogue, one must have a comprehensive and unambiguous list of acts. Despite a taxonomy of speech acts defined by Searle (1985) and subsequent refinements of speech act taxonomies motivated by philosophers of language (for example, Wallis 1995 and Schiffrin 1995), many researchers rely on their own ad-hoc acts which are often dependent on the application. A given context may rely on only a small subset of speech acts, hence researchers often choose their own to suit the application domain. Others have concentrated on building taxonomies of speech acts which are domain independent at the highest levels but become increasingly dependent to the domain at the lower levels. Alexandersson & Reithinger (1995) Alexandersson (1996) use a similar approach in the Verbmobil project, an appointment manager. A dialogue act falls into one of four broad categories: initialisation, negotiation, closing and a further category for deviation. In each category there is a set of domain independent speech acts such as accept and reject, clarify and deliberate. Each of these has domain dependent instances, for example, accept can be one of three dependent acts: accept_date, accept_duration and accept_location for an appointment. An annotated corpus then provides a statistical prediction module for the next speech act based on the previous acts.

The term ‘dialogue act’ is often used synonymously with ‘speech act’. They both covey the illocutionary force of the speaker. Often dialogue acts relate to the subset of speech acts which deal directly with dialogue management. There are often conflicting definitions of the term ‘dialogue act’; they can also combine propositional content to a speech act. For example, the act ‘accept_date’ above could contain the date being accepted. This approach has the benefit of associating the context directly with the intention.

A proposed scheme to replace speech acts is Dialogue Interpretation Theory (DIT) (Bunt 1989, 1994, 1996). The traditional view of dialogue acts is an act of communication performed in a dialogue setting. Bunt takes a more specific meaning: the acts are functional units which are used by the user to change the context, (definition used by Gazdar 1979 and Isard 1975). DIT classifies dialogue acts as either *task orientated* or for *dialogue control* and incorporate a propositional content. The scheme places each act into a hierarchy, for example dialogue control acts can fall into three more specific categories: feedback, interaction management and social obligations. The scheme is comprehensive, covering acts from turn management to topic structuring. Furthermore, social obligation acts can be used to create interactive and reactive pressures leading to discourse phenomena, such as adjacency pairs.

6.1.2 Dialogue Structure

A domain independent to dependent hierarchy can add a form of structure, but does not offer to add any more structure to a dialogue beyond the generalisation of acts. Conversational Games Theory (Carletta et al. 1995, Kowtko et al. 1991), described above (section 4.2.1), adds a form of structure to a dialogue above the speech act. Taking the view that a dialogue accomplishes a single task, the dialogue can be regarded as a series of transactions, each of which accomplish a small part of that task, i.e. a sub-task. Each transaction comprises one or more games, where each game is a pair of complementary speech acts (for example an initiating act Q-YN must have a responding REPLY-Y or REPLY-N act). Additionally, a game can be embedded in another, allowing phenomena such as clarification sequences, diversions etc. The scheme was initially developed to encode dialogue structure in a corpus of route description exercises using maps, but Williams (1996a) shows that at least the games level can be implemented to successfully predict the next spoken act.

Other attempts at deriving a structure above the speech act level include Alexandersson (1996). Taking a corpus of dialogues annotated with dialogue acts, Alexandersson can automatically acquire an intentional structure, i.e. a hierarchical structure of the dialogue, by dividing the dialogue up into goals and sub-goals. This process is threefold. First the acts in the corpus are generalised from domain dependent to independent by traversing up the dialogue act hierarchy, for example the act ‘suggest_support_date’ becomes ‘suggest’. Multiple occurrences of a generalised act are collapsed to just one. Secondly, sequences of dialogue acts are abstracted into a small set of classes, each of which performs a function or *phase* of the dialogue. This is different from generalisation since certain acts may overlap several classes. Then context-free rules are automatically generated for each class using the bayesian model merging of the Bogus System (Stolcke 1994a, 1994b). In this way, a hierarchical structure to dialogue similar to the transaction level of conversation games theory can be acquired through partial machine learning. There is a large amount of human intervention required, however, from defining the classes to deciding which acts to include in each class. Alexandersson also has reservations for the granularity of the classes and dialogue acts. Another point is that the corpus consists of dialogues which only contain one main task: making an appointment. It remains to be seen as to how effective this approach is for multiple tasks. Other applications of Machine Learning to language classification problems tend to be demonstrated on analogously limited data sets and do not generalise or scale up easily (for example, Churcher et al. 1996, Atwell 1996a, Hughes & Atwell 1994).

6.2 Recovery from errors

As mentioned above, in a dialogue the user's task may fail, but the dialogue should never. A dialogue manager should be able to detect errors or misunderstandings and recover from them or at the very least fail gracefully. Errors can occur from many quarters, from the speech recogniser to a user's misinterpretation. To be able to recover from an error, the manager needs to be able to identify it and employ a suitable strategy. The strategy may differ depending on the type of error and needs to be recursive: when attempting to correct an error, more may arise. Luperfoy (1996) employs a recovery strategy according to a four stage algorithm:

- **Detection** - detecting the error can be quite difficult and may rely on the user notifying the system of a problem
- **Diagnosis** - classification of error
- **Repair plan selection** - according to the class of error
- **Interactive plan execution** - recovery plan is executed interactively so that any errors which arise from the plan can be dealt with.

Luperfoy identifies eight classes of error from the speech recogniser not receiving any input to errors occurring in the application. Other researchers, for example, Taleb (1996) and Veldhuijzen van Zanten (1996) identify errors which depend more on the technology. For example, Veldhuijzen van Zanten identifies three main types of error which are typical of those generated by speech recognition:

- Uncertainties - the speech recogniser has low confidence in the speech it has received.
- Inconsistencies - a misunderstanding; the speech may have a high confidence value but conflicts either with the domain model or previous conversation.
- Ambiguities - there is more than one interpretation of the speech, each having a high confidence.

The chosen strategy will depend on the type of error. For example, if the manager has low confidence in what has been said, it may ask the user for confirmation or to repeat. Ambiguities can be dealt with in a similar way; if two words sound similar and the manager receives a high confidence for each, e.g. "plane" and "train", rather than asking the user to repeat him/herself hence perpetuating the problem, the manager could ask to confirm one of the words: "Did you say train?" to which the reply would be either "yes" or "no, plane".

By modelling the domain of the application, inconsistencies can be detected before being passed to the application itself. A good use of this is demonstrated by Luperfoy who distinguishes between training and tutoring in an army forces simulator, ModSAF. When a user is using the system under tutoring, the dialogue manager can correct the user's mistakes before passing them onto the application. The corrections can be justified to the user allowing him/her to correct his/her mistake without disastrous consequences to the simulation. Without the domain model this would not be possible and the manager will have to interpret any error messages generated by the application.

In Taleb (1996), the author presents an analysis of communication deviations, that is where an error has occurred and the dialogue changes to correct the error. By examining the types of misunderstandings which arise in human to human conversation and counting the number of turns it takes to recover from them, an understanding of the frequency and cost of different types of error can be built up. Taleb suggests two categories of deviation: content and role deviation. Content deviations can consist of ambiguous questions where a question can be interpreted in more than one way and a disambiguating question is asked (e.g. "Did you say train?"), fuzzy questions where there are multiple interpretations and clarification is required, knowledge which is either incorrect or not shared. Role deviations include initiative reversing where the person who is usually answering asks a question or takes the initiative in some other way, evaluation of skill, identification of role where the person who is responsible for different kinds of information is established and an evaluation of relevance.

Taleb found that to the exclusion of ambiguous questions which occurred infrequently and took few turns to recover from, the more frequent the error the fewer number of turns it took to correct. Role deviations were the most frequent, especially initiative reversing, although evaluation of skill occurred the least often. The corpus of analysis was between two humans but parallels can be drawn for human to machine dialogues. Although a similar classification could be used, it remains to be seen whether the types of errors occur with the same frequencies. The problems inherent in speech recognition for example, may lead to more content deviations as misunderstandings are more likely.

6.3 Design of System Prompts

The design of system prompts, as mentioned above in sections 3.3.1 and 5, can greatly influence the user's language. Prompt design is important for natural, flowing conversation, to overcome shortcomings in speech recognition technology and the transition of graphical user interfaces to speech interfaces. One of the most challenging aspects of prompt design is implicitly letting the user know what they can say. By not knowing, users can go beyond the functionality of the system, or conversely, not utilise the system as fully as they could.

For a user using the system for the first time, it may not be obvious what can actually be said. The option of explicitly stating all of the options open to the user in a prompt may prove annoying due to the slow communication rate of speech. Yankelovich (1996) suggests a continuum between implicit and explicit prompting which can be tailored to the user. Implicit prompts are often open-ended leaving it up to the user to make a choice in what they say and how they say it. For example, "This is AZ Banking. What would you like to do?". This degree of flexibility may be preferable for the user, but leads to large, ambiguous grammars with high recognition error rates.

At the other end of the scale, explicit prompts typically tell the user what can be said for each option. For example, "This is AZ Banking. Say 'check balance' to check your balance, 'pay bill' to pay a bill, or 'transfer funds' to transfer funds between accounts." Prompts designed in this way can be tediously long, and do not allow the user to express themselves as they wish. On the other hand, they constrain the user's response, leading to far better accuracy than their implicit counterparts.

Yankelovich suggests making prompts more explicit in the case of recognition errors and less explicit as the user shows greater familiarity with the system. By starting with system prompts which fall between the two extremes, users can discover how to use the system whilst using language which is natural to them. If the user does not respond to the prompt or his/her response cannot be interpreted, then the prompt can be made incrementally more explicit. For example,

System: This is AZ Banking. You can check your balance, pay a bill or transfer funds between accounts.
What would you like to do?
User: <recognition error>
System: To check your balance, say 'check balance', to pay a bill, say 'pay bill', to transfer funds between
accounts, say 'transfer funds'.

Making prompts less explicit when the user shows a degree of familiarity with the system can speed up interaction and make it appear to flow more naturally. Yankelovich calls this 'tapering'. The user is at first presented with an explicit prompt. When the user comes across the prompt again, perhaps by making another request, the prompt is shortened. Another method of shortening the prompts as the user gains experience is to introduce *hints*. Prompts comprise an implicit part followed by an explicit example of a response. As the user gains familiarity with the system, the explicit hint can be removed. Although the prompt may now appear open-ended, the user will respond in an anticipated manner.

There appears to be a trade-off between the user's flexibility in responding to a prompt and speech recognition accuracy. By using the effects of priming in a positive manner, it may be possible to minimise this. The expansion of prompts as a means of recovery can compensate for cases where this is not as effective. Progressive assistance is a more natural, and certainly more useful method of recovery than the often and ill-used prompt, "Please repeat."

7. Evaluation of a DMS

Evaluating a dialogue management system can be a difficult process depending on which aspect is evaluated. The robustness of the system, that is whether it can recover from different types of error, can be objectively measured. Testing the system's robustness should be part of the design stage. But how can we tell whether the system manages dialogue in an effective and even pleasant way? How can you measure pleasantness? There are two main approaches to the evaluation of dialogue managers: the qualitative and quantitative views.

7.1 Qualitative evaluation

A qualitative evaluation would rely on the user's opinion of the system. Dybkjaer et al. (1995) conducted interviews after the user used the system and asked whether the dialogue seemed natural and pleasant. This subjective assessment may seem fair, but is fraught with problems. For example, the user may learn after the first attempt how to address the system, which words to use or not to use. Subsequent uses of the system may produce different evaluations even though the system has not changed. Some users may find the system difficult to use whilst others will find it effortless. 'Pleasantness' differs from person to person, too.

As Vilnat (1996) argues, there is no consensus of what comprises a good dialogue. When asking the user, the designer has to make sure that the user is representative of the end user in terms of background and frequency of use. The designer needs to conduct many evaluations to take into account any variations between users. Because of these problems, many researchers have tried to provide a means of objectively evaluating a system.

7.2 Quantitative evaluation

There are two methodologies for quantitatively evaluating a dialogue management system: black box and glass box. The black box method is concerned with input and output behaviour, whilst the glass box method evaluates the behaviour of individual components in the system. Glass box evaluation can rely on a comparison between the output of a component and a retrospective reference. By directly comparing the two it is possible to measure the accuracy of that component. The black box approach, on the other hand, cannot use this method to evaluate a dialogue since there is no 'correct' dialogue to compare it with. Despite this, objective evaluation of the dialogue is necessary in order to compare the performance of different systems. Initial efforts have been made to standardise this (see EAGLES 1994, Fraser 1995b) but remain work in progress.

Fraser (1995a) and Eckert (1996) derive a measure of its success from the number of turns taken in a dialogue . This is problematic, however, since the dialogue may not end satisfactorily for the user. It also favours terse, efficient dialogues which may seem unnatural to the user. As Fraser points out, the majority of language is "messy" and hence a certain amount of inefficiency may make the dialogue pleasant. The number of turns taken by a dialogue cannot be used to make direct comparisons between systems. Fraser also notes that in some dialogues there is a degree of repetition; user's become ensnared in loops where the system continually makes the same mistake. The proportion of the number of turns spent in repetition may be a more useful comparison, but it is often difficult to detect when a user is repeating parts of a dialogue. Rather than simply detecting whether the system has been in the same state before, overall patterns of behaviour need to be detected.

A more analytical approach needs to be taken. Taleb's (1996) communication deviations (see section 6.2, above) could be utilised for this purpose. The frequency of differing classes of errors and the cost associated with their recovery can be used to compare system's performance. If a benchmark standard for differing types of dialogue systems is produced, then a system's performance can be measured. For an accurate comparison, however, there needs to be a standard taxonomy for describing a dialogue system. Until this is achieved then direct comparisons between systems mean little.

Churcher et al. (1997) proposes the use of a generic template for the evaluation of dialogue management systems, based on the model in Figure 2. The template provides an application-independent method for assessing systems according to the features they exhibit. Rather than thinking of evaluation in terms of speed/accuracy of analysis of some standard test corpus, dissimilar, rival DMSs from different domains can be meaningfully compared by assessing how well they match this generic template for dialogue management architecture. This "genericness" score can temper any measures of speed, accuracy, naturalness etc., in much the same way as Atwell (1996b) proposes by the evaluation of parsing schemes against a generic EAGLES-derived template of syntactic levels.

8. Conclusions

This report has provided an overview of some of the differing issues affecting dialogue management systems. Their primary use is envisaged as a spoken language front end to an application. Rather than relying on one-shot question and answer style interaction, a dialogue is a more personal affair establishing the exact requirements from a user. As a result, the evaluation of the system becomes a subjective one, despite attempts to produce quantitative measures. The precise role of dialogue management systems remains to be seen. Although there have been a number of successfully implemented systems, a recent survey of user's opinions by Dybkjaer et al. (1995) showed that most preferred talking to a human. Whether this reflects a lack of pleasantness which has yet to be implemented in a system, or the user's lack of willingness to use a new technology remains to be seen. Vilnat (1996) favours the subjective assessment of systems, after all, the user is the best person to carry out an evaluation and the simplest question you can ask him or her is, would they like to use it again?

Bibliography

Ahrenberg et al.
L. Ahrenberg, N. Dahlbäck and A. Jönsson (forthcoming), “Dialogue management for natural language interfaces”, Manuscript in preparation.

Alexandersson & Reithinger 1995
J. Alexandersson and N. Reithinger, “Designing the dialogue component in a speech translation system - a corpus-based approach”, in Andernach et al. (eds.) 1995.

Alexandersson 1996
J. Alexandersson, “Some ideas for the automatic acquisition of dialogue structure”, in Luperfoy et al. 1996.

Allen 1991
J. F. Allen, “Discourse structure in the TRAINS project”, Proceedings of the 4th DARPA Workshop on Speech and Natural Language. 1991.

Andernach et al. 1995
J. A. Andernach, S. P. van de Burgt and G. F. van der Hoeven (eds), “Corpus-based Approaches to Dialogue Modelling”, Proceedings of the 9th Twente Workshop on Language Technology , University of Twente, Enschede, Netherlands. 1995.

Anderson et al. 1991
A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty-Sneddon, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson and R. Weinert, “The HCRC Map Task Corpus”, Language and Speech, 34(4), pp 351-366. 1991.

Atwell 1996a
E. S. Atwell, “Machine learning from corpus resources for speech and handwriting recognition”, in: Thomas, J & Short, M (editors) Using Corpora for Language Research: Studies in Honour of Geoffrey Leech, pp.151-166. Longman, 1996.

Atwell 1996b
E. S. Atwell, "Comparative evaluation of grammatical annotation models", to appear in: R. Sutcliffe, H. D. Koch & A. McElligott (eds.), "Industrial Parsing of Software Manuals", Rodopi. 1996.

Aust & Oerder 1995
H. Aust and M. Oerder, “Dialogue control in automatic inquiry systems”, in Andernach et al. (eds.) 1995.

Austin 1962
J. L. Austin, “How to do things with words”, Oxford University Press, London. 1962.

Baekgaard et al. 1995
A. Baekgaard, N. O. Bernsen, T. Broendsted, P. Dalsgaard, H. Dybkjær, L, Dybkjær, J. Kristiansen, L. B. Larsen, B. Lindberg, B. Maegaard, B. Music, L. Offersgaard, C. Povlsen, “The Danish Spoken Language Project - A general overview”, Proceedings of ETRW on Spoken Dialogue Systems, Vigs. 1995.

Belina & Hogrefe 1988
F. Belina and D. Hogrefe, “The CCITT Specification and Description Language SDL”, in Computer Networks and ISDN Systems 16, pp 311-341, N. Holland. 1988.

Bernsen et al. 1995	N. O. Bernsen, H. Dybkjær, L, Dybkjær, “Scenario Design for Spoken Language Dialogue Systems Development”, Proceedings ETRW on Spoken Dialogue Systems, Vigs, pp 93-96. 1995.
Beun 1996	R. J. Beun, “Speech Act generation in cooperative dialogue”, in Luperfoy et al. 1996.
Bird et al. 1995	S. Bird, S. Browning, R. Moore and M. Russell, “Dialogue move recognition using topic spotting techniques”, ESCA Workshop on Spoken Dialogue Systems, Denmark. 1995.
Bratman et al. 1988	M. E. Bratman, D. J. Israel and M E. Pollack, “Plans and resource-bounded practical reasoning”, Computational Intelligence, 4, pp 349-355. 1988.
Brown & Yule 1983	G. Brown and G. Yule, “Discourse analysis”, Cambridge Textbooks in Linguistics, Cambridge: Cambridge University Press. 1983.
Bunt 1989	H. C. Bunt, “Information Dialogues as Communicative Actions in Relation to User Modelling and Information Processing”, in Taylor et al. (1989). 1989.
Bunt 1994	H. C. Bunt, “Context and Dialogue Control”, THINK Quarterly 3(1), 19-31. 1994.
Bunt 1996	H. Bunt, “Interaction management functions and context representation requirements”, in Luperfoy et al. 1996.
Carletta et al.	J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon and A. Anderson (in preparation), “Coding dialogue structure in the map task Ms.”. In preparation for journal submission.
Carletta et al. 1995	J. H. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon and A. Anderson, “The coding of dialogue structure in a corpus”, in Andernach et al. (eds.) 1995.
Chu-Carroll 1996	J. Chu-Carroll, “Response generation in collaborative dialogue interactions”, in Luperfoy et al. 1996.
Churcher et al. 1996	G. E. Churcher, D. C. Souter, E. S. Atwell, “Dialogues in Air Traffic Control”, in Luperfoy et al. 1996.
Churcher et al. 1997	G. E. Churcher, D. C. Souter, E. S. Atwell, "Generic Template for the evaluation of Dialogue Management Systems", submitted to EUROSPEECH'97, 1997.
Clark & Schaefer 1989	H. Clark and E. Schaefer, “Contributing to Discourse”, Cognitive Science, 13, pp 259-294. 1989.
Clark 1985	H. H. Clark, “Language use and language users”, in Lindzey & Aronson (1985). 1985.

Cohen 1996 P. Cohen, “Dialogue Modelling”, in R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, V. Zue, “Survey of the State of the Art in Human Language Technology”, chapter 6.3. 1996. On WWW at http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html

Cole & Morgan 1975 P. Cole and P. J. Morgan (eds.), “Syntax and Semantics 3: Speech Acts”, pp 41-58, Academic Press, Inc. 1975.

Connolly et al. 1995 J. H. Connolly, A. A. Clarke, S. W. Garner and H. K. Palmén, “Clause-internal structure in spoken dialogue”, in Andernach et al. (eds.) 1995.

Coulthard & Brazil 1992 M. Coulthard and D. Brazil, “Exchange Structure”, in Coulthard (1992). 1992.

Coulthard 1992 M. Coulthard (ed.), “Advances in spoken discourse”, London: Routledge, pp50-78. 1992.

Dahlbäck 1995 N. Dahlbäck, “Kinds of agents and types of dialogues”, in Andernach et al. (eds.) 1995.

Dalsgaard & Baekgaard 1994 P. Dalsgaard and A. Baekgaard, “Spoken Language Dialogue Systems”, in Freksa (1994) pp 178-191, 1994.

Dalsgaard et al. 1995 P. Dalsgaard et al. (eds.), Proceedings of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems, Vigsø, Denmark. 1995.

Dybkjær & Dybkjær 1993 L. Dybkjær and H. Dybkjær, “Wizard of Oz Experiments in the development of the dialogue model for P1”, Technical Report 3, Center for Cognitive Informatics, Roskilde University. 1993.

Dybkjær et al. 1995 H. Dybkjær, L. Dybkjær, N. O. Bernsen, “Design, formalization and evaluation of spoke language dialogue”, in Andernach et al. (eds.) 1995.

EAGLES 1994 EAGLES: Spoken Language Systems. DRAFT- work in progress, Oct. 1994.

Eckert 1996 W. Eckert, “Understanding of spontaneous utterances in human-machine dialogue”, in Luperfoy et al. 1996.

Ervin-Tripp 1979 S. Ervin-Tripp, “Children’s verbal turn-taking”, in Ochs & Schieffelin (1979: 391-414). 1979.

Fraser & Gilbert 1991 N. M. Fraser and G. N. Gilbert, “Simulating Speech Systems”, Computer Speech and Language 5, pp 81-99, 1991.

Fraser 1995a N. Fraser, “Messy data, what can we learn from it?”, in Andernach et al. (eds.) 1995.

Fraser 1995b N. Fraser, “Quality Standards for Spoken Dialogue Systems: a report on progress in EAGLES”, in Dalsgaard et al. (1995), pp 157-160. 1995.

Freeman 1992 — D. Freeman, “Collaboration: constructing shared understandings in a second language classroom”, in Nunan (1992). 1992.

Freksa 1994 — C. Freksa, “Prospects and Perspective in Speech Technology”, Proceedings in Artificial Intelligence, Infix, München, Germany. September 1994.

Gazdar 1979 — G. R. Gazdar, “Speech Act assignment”, in Joshi et al. (1979). 1979.

Grice 1975 — H. P. Grice, “Logic and Conversation” in Cole & Morgan (1975). 1975.

Grosz & Kraus 1993 — B. J. Grosz and S. Kraus, “Collaborative plans for group activities”, in Proceedings of the IJCAI. 1993.

Grosz & Sidner 1990 — B. J. Grosz and C. L. Sidner, “Plans for discourse”, in Cohen, Morgan & Pollack (eds.), Intentions in Communication, chapter 20, pp 417-444. MIT Press. 1990.

Grosz 1977 — B. J. Grosz, “The representation and uses of focus in dialogue”, PhD thesis, University of California, Berkeley. 1977.

Guindon 1988a — R. Guindon, “A multidisciplinary perspective on dialogue structure in user-advisory dialogues”, in Guindon (1988b). 1988.

Guindon 1988b — R. Guindon (ed.), “Cognitive Science and its Application for Human-Computer Interaction”, Lawrence Erlbaum Publishers. 1988.

Hughes & Atwell 1994 — J. Hughes, E. S. Atwell, "A Methodical Approach to Word Class Formation Using Automatic Evaluation", in Lindsay Evett and Tony Rose (eds), Proceedings of AISB workshop on Computational Linguistics for Speech and Handwriting Recognition, Leeds University. 1994.

Hulstijn et al. 1996 — J. Hulstijn, R. Steetskamp, H. ter Doest, S. van de Burgt and A. Nijholt, “Topics in SCHISMA Dialogues”, in Luperfoy et al. 1996.

Isard 1975 — S. Isard, “Changing the context”, in Keenan (1975). 1975.

Jefferson 1972 — G. Jefferson, “Side Sequences”, in Sudnow (1972: 294-338). 1972.

Jokinen 1995 — K. Jokinen, “Goal formulation based on communicative strategies”, in Proceedings of the 16th COLING, 1996.

Jokinen 1996 — K. Jokinen, “Cooperative response planning in CDM: Reasoning about communicative strategies”, in Luperfoy et al. 1996.

Jönsson 1993 — A. Jönsson, “Dialogue Management for Natural Language Interfaces”, PhD thesis, Linköping University. 1993.

Joshi et al. 1979 — A. K. Joshi, I. A. Sag and B. L. Webber (eds.), “Elements of discourse understanding”, Cambridge University Press, Cambridge, UK. 1979.

Keenan 1975 — E. L. Keenan, “Formal semantics of natural language”, Cambridge University Press, Cambridge, UK. 1975.

Kennedy et al. 1988 — A. Kennedy, A. Wilkes, L. Elder and W. Murray, “Dialogue with machines”, Cognition, 30, pp 73-105. 1988.

Kowtko et al. 1991 — J. C. Kowtko, S. D. Isard, G. M. Doherty, “Conversational Games within Dialogue”, in Proceedings of the ESPRIT Workshop on Discourse Coherence, University of Edinburgh, April 1991.

Larsen 1996 — L. B. Larsen, “Voice Controlled home banking - objectives and experiences of the ESPRIT OVID project”, in Proceedings of the 3rd IEEE workshop on Interactive Voice Technology for Telecommunications Applications”, New Jersey USA, September 1996.

Leech 1983 — G. N. Leech, “Principles of Pragmatics”, Longman Linguistics Library. 1983.

Levinson 1981 — S. Levinson, “Some pre-observations on the modelling of dialogue”, Discourse Processes”, 4(1), pp 93-116. 1981.

Levinson 1983 — S. C. Levinson, “Pragmatics”, Cambridge Textbooks in Linguistics, Cambridge: Cambridge University Press. 1983.

Lewin et al. 1994 — I. Lewin, M. Russell, D. Carter, S. Browning, K. Ponting and S. Pulman, “Conversational Games Within Dialogue”, Human Communication Research Centre, Edinburgh University. 1994.

Lindzey & Aronson 1985 — G. Lindzey and E. Aronson (eds.), “The Handbook of Social Psychology”, 3rd Edition, Erlbaum. 1985.

Löbner 1996 — H. Löbner, Löbner Prize Home Page, at http://acm.org/~loebner/loebner-prize.htmlx

Luperfoy 1996 — S. Luperfoy, “Tutoring versus Training: A mediating dialogue manager for spoken language systems”, in Luperfoy et al. 1996.

Luperfoy et al. 1996 — S. Luperfoy, A. Nijholt and G. Veldhuijzen van Zanten (eds.), “Dialogue Management in Natural Language Systems”, Proceedings of the 11th Twente Workshop on Language Technology , University of Twente, Enschede, Netherlands. 1996.

MacDermid 1993 — C. MacDermid, “Features of Naïve Callers’ Dialogues with a Simulated Speech Understanding and Dialogue System, in EUROSPEECH 93, 15, pp 955-958. 1993.

Maier 1996 — E. Maier, “Context construction as subtask of dialogue processing – the VERBMOBIL case”, in Luperfoy et al. 1996.

McGlashan 1996 — S. McGlashan, “Towards multimodal dialogue management”, in Luperfoy et al. 1996.

McKevitt et al. 1992 — P. McKevitt, D. Partridge and Y. Wilks, “Approaches to natural language discourse processing”, in Artificial Intelligence Review, Vol. 6, No. 4, 1992.

Moore 1995 — J. D. Moore, “Participating in explanatory dialogues. Interpreting and responding to questions in context”, MIT Press, Cambridge, Mass., 1995.

Müller & Runger 1993 — C. Müller and F. Runger, “Dialogue Design Principles - Key for usability of voice processing”, in Proceedings of the 3rd European Conference on Speech, Communication and Technology (EUROSPEECH 93), pp. 943-946, Berlin, Germany, 1993.

Nielsen & Baekgaard 1992 — P. B. Nielsen and A. Baekgaard, “Experience with a dialogue description formalism for realistic applications”, in Proceedings of the International Conference on Spoken Language Processing (ICSLP 92), pp 719-722, Banff, Canada, 1992.

Novick & Hansen 1995 — D. G. Novick and B. Hansen, “Mutuality strategies for reference in task-orientated dialogue”, in Andernach et al. (eds.) 1995.

Novick & Sutton 1996 — D. G. Novick and S. Sutton, “Building on experience: managing spoken interaction through library subdialogues”, in Luperfoy et al. 1996.

Novick & Ward 1993 — D. G. Novick and K. Ward, “Mutual Beliefs of Multiple Conversants: A computational model of collaboration in Air Traffic Control”, in Proceedings of AAAI’93. 1993.

Nugues et al. 1996 — P. Nugues, C. Godéreaux, P. El Guedj and F. Revolta, “A conversational agent to navigate in virtual worlds”, in Luperfoy et al. 1996.

Nunan 1992 — D. Nunan (ed.), “Collaborative Language Learning and Teaching”, Cambridge: Cambridge University Press. 1992.

Nunan 1993 — D. Nunan, “Introducing Discourse Analysis”, Penguin English Applied Linguistics, London: Penguin. 1993.

Ochs & Schieffelin 1979 — E. Ochs and B. B. Schieffelin (eds.), “Developmental Pragmatics”, New York: Academic Press. 1979.

Rubin 1980 — A. Rubin, “A theoretical taxonomy of the differences between oral and written language”, in Spiro et al. (1980). 1980.

Sacks et al. 1974 — H. Sacks, E. A. Schegloff and G. Jefferson, “A simplest systematics for the organization of turn-taking in conversation”, Language, 50.4, 696-735. 1974.

Schegloff & Sacks 1973 — E. A. Schegloff & H. Sacks, “Opening up closings”, Semiotica, 7.4, 289-327. 1973.

Schiffrin 1995 A. Schiffrin, “A computational treatment of conversational contexts in the theory of speech acts”, MSc Thesis, School of Computer Studies, Leeds University, Leeds. 1995.

Schillo 1996 M. Schillo, “Working while Driving: Corpus-based language modelling of a natural English voice-user interface to the in-car personal assistant”, MSc Thesis, School of Computer Studies, Leeds University, Leeds, UK. 1996.

Searle & Vanderveken 1985 J. R. Searle, D. Vanderveken, “Foundations of Illocutionary Logic”, Cambridge University Press, Cambridge. 1985.

Searle 1969 J. R. Searle, “Speech Acts: An Essay in the Philosophy of Language”, Cambridge University Press, Cambridge. 1969.

Searle 1975 J. R. Searle, “Indirect Speech Acts”, in Syntax and Semantics, Vol. 3: Speech Acts, ed. J. L. Morgan, Academic Press, New York. 1975.

Shieber 1994 S. M. Shieber, “Lessons from a restricted Turing Test”, Computation and Language archive, at http://xxx.lanl.gov/abs/comp-lg/9404002. 1994

Sidner 1992 C. L. Sidner, “Using discourse to negotiate in collaborative activity: an artificial language”, in AAAI-92 Workshop, “Cooperation among heterogeneous intelligent systems”, pp 121-128. 1992.

Sidner 1994 C. L. Sidner, “An artificial discourse language for collaborative negotiation”, in Proceedings of the AAAI, pp 814-819. 1994.

Sinclair & Coulthard 1975 J. McH. Sinclair & M. Coulthard, “Towards an analysis of discourse: the English used by teachers and pupils”, Oxford: Oxford University Press. 1975.

Spiro et al. 1980 R. J. Spiro, B. B. Bruce and W. F. Brewer (eds.), “Theoretical Issues in Reading Comprehension”, Erlbaum. 1980.

Stolcke 1994a A. Stolcke, “Bayesian Learning of Probabilistic Language Models”, PhD thesis, University of California, Berkley. 1994.

Stolcke 1994b A. Stolcke, “How to Boogie: A Manual for Bayesian Object-orientated Grammar Induction and Estimation”, International Computer Science Institute of Berkley, California. 1994.

Sudnow 1972 D. Sudnow (ed.), “Studies in social interaction”, New York: The Free Press. 1972.

Taleb 1996 L. Taleb, “Communicational deviation in finalized informative dialogue management”, in Luperfoy et al. 1996.

Taylor et al. 1989 M. M. Taylor, F. Néel and D. G. Bouwhuis (eds.),“The Structure of Multimodal Dialogue”, Amsterdam: North Holland Elsevier. 1989.

Traum & Hinkelman 1992 D. R. Traum and E. Hinkelman, “Conversation acts in task-orientated spoken dialogue”, Technical Report No. 425, Computer Science Department, University of Rochester, USA (also in Computational Intelligence, 8(3), August 1992).

Traum 1991 D. R. Traum, “Towards a computational theory of grounding in natural language conversation”, Technical Report No. 401, Computer Science Department, University of Rochester, USA. 1991.

Traum 1996 D. R. Traum, “Conversational Agency: The TRAINS-93 Dialogue Manager”, in Luperfoy et al. 1996.

Turing 1950 A. M. Turing, “Computer Machinery and Intelligence”, Mind LIX No. 236. 1950.

Veldhuijzen van Zanten 1996 G. Veldhuijzen van Zanten, “Pragmatic interpretation and dialogue management in spoken-language systems”, in Luperfoy et al. 1996.

Vilnat 1996 A. Vilnat, “Which processes to manage human-machine dialogue?”, in Luperfoy et al. 1996.

Wallis 1995 H. Wallis, “Mood, speech acts and context”, PhD Thesis, Department of Philosophy, Leeds University, Leeds. 1995.

Weizenbaum 1966 J. Weizenbaum, “Eliza”, Communications of the ACM 9, pp36-45. 1966.

Widdowson 1978 H. G. Widdowson, “Teaching language as communication”, Oxford: Oxford University Press. 1978.

Williams 1996a S. Williams, “Dialogue Management in a mixed-initiative, co-operative, spoken language system”, in Luperfoy et al. 1996.

Williams 1996b S. H. Williams, “Anaphoric reference and ellipsis resolution in a telephone-based spoken language system for accessing email”, DAARC’96. 1996.

Yankelovich 1996 N. Yankelovich, “How do users know what to say?”, ACM Interactions, Vol. 3, No. 6, 1996.