Planning Efficient Mixed Initiative Dialogue

Eli Hagen*
Deutsche Telekom, FZ131
PO Box 10 00 03
64276 Darmstadt, Germany
email: hagen@fz.telekom.de

Brigitte Grote
Dept. of Computer Science
Otto-von-Guericke Universität Magdeburg
39106 Magdeburg, Germany
email: grote@iik.cs.uni-magdeburg.de

1 Motivation

A common feature of a number of current spoken dialogue systems for information retrieval is that little emphasis is placed on the generation of system contributions to the dialogue. In these systems, utterances have mostly been produced from templates, see for instance (Whittaker and Attwater 1994; Blomberg *et al.* 1993; Oerder and Aust 1993; Meng *et al.* 1996). This is a valid approach in system initiative type systems and in systems where utterances stand in a one-to-one relation to communicative goals. In mixed initiative systems, however, user and system might both lead the dialogue by providing several pieces of information and pursuing several different goals within one utterance. Hence, in this kind of dialogue we cannot predict what information the user chooses to provide, and hence cannot predict the system's response. We argue that in any system of reasonable size, the number of templates would be too large to determine *a priori*. Instead, in order to achieve efficient and cooperative dialogue, system utterances must be generated using natural language generation (NLG) techniques.

NLG has been used by, for instance, (Pan and McKeown 1996; Sadek *et al.* 1996), but they put emphasis on generation of system answers, i.e. on offering and providing information. We are concerned with utterances requesting specific information. (Sadek *et al.* 1996) also generate requests, but they are mostly of a general nature, and the sample dialogue in their paper suggests that the system's requests for specific information can only realize one communicative goal at a time.

In several recent systems (e.g., (Allen *et al.* 1996; Sadek *et al.* 1996)), confirmation of information that the system acquires from the user (variables) is mostly avoided. This might be a valid approach as their number of variables is small, but we think that it will lead to frustrating dialogues when several variables are needed. Consider the following interaction, where the system actually misrecognized all the values provided by the user. User: "How much is a call from Bonn to Homburg at 8 o'clock?" Sys: "A call from Berlin to Hamburg at 1800 costs 36 Pfennig per minute." It would be very hard to recover from this situation and the user would most likely never call again.

*The author is a visiting researcher in the speech processing group FZ131, Deutsche Telekom. Authors are in opposite alphabetical order this time.

In our domain (calculation of the price of phone calls), the system must acquire several variables—with sometimes ambiguous values. Therefore confirmation and disambiguation questions are necessary, and hence we have a larger number of communicative goals to satisfy than the afore mentioned systems. But, if we were to designate a single utterance for each communicative goal, we would quickly end up with inefficient and annoying dialogues like the following:

Sample dialogue 1
A. Sys: Do you want the rate or the total cost of a call?
B. Usr: How much is a call to Frankfurt, please.
C. Sys: Do you want the rate or the total cost of a call?
D. Usr: The rate of a call to Frankfurt.
E. Sys: Do you want the rate of a call?
F. Usr: Yes, to Frankfurt, please.
G. Sys: What is the destination of your call?
H. Usr: Frankfurt!
I. Sys: Is the destination of your call Frankfurt?
J. Usr: Yess!!
K. Sys: Frankfurt am Main or Frankfurt an der Oder?
L. Usr: Frankfurt am MAAAAIIIN!!!

To avoid such uncooperative dialogues, a system has to be able to interpret additional information provided by the user, as, for instance, in move D in dialogue 1. Further, system responses must be efficient. Humans achieve efficiency by pursuing several goals at a time instead of dealing with single goals in a strict sequential order. In our approach, we apply this observation to the design of information systems, hoping that it results in an interaction as illustrated in dialogue 2. Here, some goals are expressed implicitly (e.g., confirmation in utterance C), while others are omitted (e.g., asking for the destination).

Sample dialogue 2
A. Sys: Do you want the rate or the total cost of a call?
B. Usr: How much is a call to Frankfurt, please.
C. Sys: The rate or the total cost of a call to Frankfurt?
D. Usr: The rate, please.
E Sys: Frankfurt am Main or Frankfurt an der Oder?
D. Usr: Am Main.

In this paper, we describe an initial realization of such a cooperative and efficient mixed initiative dialogue system. In particular, we discuss system utterances whose primary goal is to acquire information of various kinds, since these occur frequently in our domain. Building on results in (Hagen 1997), we develop heuristics for jointly expressing several communicative goals in one utterance, thus responding to the requirements of the task at hand and to the user initiative at the same time. A prototypical system that answers

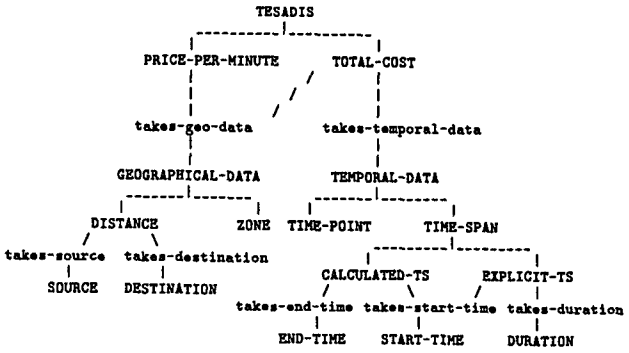

Figure 1: An abridged version of the task description for the TESADIS telephone rate inquiry system.

queries about the cost of telephone calls is currently being implemented.

2 Knowledge sources

We assume the existence of three different knowledge sources: A task model describing the tasks a system can perform, a model of information seeking dialogue, and a dialogue history.

The task description defines the information units that can be negotiated between the participants of a dialogue. As such, it facilitates the choice of a topic for system utterances and provides expectations regarding potential user responses. Negotiated topics can be either alternative ways of solving a task or pieces of information needed to solve a given task. Tasks are organized in hierarchies of concepts and relations between them. A part of the task model for our application is given in Figure 1. CAPITAL LETTERs name concepts; unnamed relations denote subconcept relations; named relations represent particular relations holding between concepts (e.g., `takes-temporal-data`).

Subconcept relations imply that there exist different ways of accomplishing a task, for instance: TESADIS can calculate either PRICE-PER-MINUTE or TOTAL-COST of a call. Concepts that participate as the range in a named relation represent obligatory subtasks. For instance, in order to calculate TOTAL-COST, the system needs information on the locational and temporal setting of the call, indicated by the relations `takes-geo-data` and `takes-temporal-data` between TOTAL-COST and GEOGRAPHICAL-DATA and TEMPORAL-DATA.

Dialogue model Our model of information seeking dialogue is speech-act oriented and a simplified version of the dialogue grammar is presented in Figure 2 (see (Sitter and Stein 1992) for a detailed discussion). Each constituent has two parameters—information seeker and information provider. The first parameter represents the initiator, or speaker, the second the hearer. During execution, the parameters are instantiated to either 'user' or 'system'. Moves in square brackets ([]) are optional and X^+ means one or more instances of constituent X.

Dialogue history In our system, the dialogue model is used analytically to build a parse tree of the dialogue with respect to the dialogue grammar. It is

```
Dialogue(S,K) → (Cycle(S,K))+
Cycle(S,K) → Request(S,K),Promise(K,S),Inform(K,S),Evaluate(S,K).
Cycle(S,K) → Offer(K,S),Accept(S,K),Inform(K,S),Evaluate(S,K).
Cycle(S,K) → Offer(K,S),[Accept(S,K)],WithdrawOffer(K,S).
Cycle(S,K) → Offer(K,S),Accept(S,K),WithdrawAccept(S,K).
Request(S,K).
Request(S,K) → request(S,K),[Dialogue(K,S)].
Request(S,K) → request(S,K),[Assert(S,K)].
Request(S,K) → Dialogue(K,S).
Request(S,K) → Assert(S,K),[request(S,K)].
Request(S,K) → Assert(S,K),[Dialogue(K,S)].
request(S,K).
Inform(K,S) → inform(K,S),[Dialogue(S,K)].
...
```

Figure 2: A simplified grammar representation of the dialogue model in pseudo-Prolog syntax.

used generatively to predict what can happen next in a dialogue. If one of the dialogue partners provides more than one dialogue act in one turn, the acts are reflected in the parse tree as several partially finished sub-trees, i.e., there are several open ends from which the dialogue might continue. Figure 3 shows a dialogue history with three open ends labelled 1–3 (see (Hagen 1997) for further discussion)

In addition to the parse tree, we consider the state of the task model part of the dialogue history. Individual nodes (i.e. concepts) in the task model can be in one of several states: open/closed nodes are nodes for which the system has not/has acquired a confirmed value. Nodes that are still under consideration, i.e., the system has requested a value or the user has provided a value, are in state topic. Topic is further divided into ambiguous, misrecognition, and confirm. confirm means that the system has low confidence in its recognition result, ambiguous means that the system has discovered that a value is ambiguous, and misrecognition means that a value has been wrongly understood. Initially, all nodes are open. During the course of a dialogue, transformations to other states take place depending on the quality of the acquired data.

3 Planning and Interpretation of dialogue continuations

The dialogue model provides all possible continuations of a dialogue, while the dialogue history defines the context in which to calculate the continuations. The parse tree contains several open ends that might serve as starting points for further dialogue contributions, and the state of the task model defines which of these are still relevant before the continuations are calculated. The continuations are represented as partial trees, and those chosen extends the parse tree at the appropriate open end. Consider the following dialogue fragment:

A. Sys: Where do you want to call? (request)
B. Usr: I want to call Hamburg. (inform)

Part (a) of Figure 3 shows the corresponding parse tree, while tree (b) shows a possible continuation, which could result in the utterance "Did you say Hamburg?" if chosen as the continuation of Inform(u,s) and if the recognition rate for Hamburg is low.

By choosing a particular continuation of the dialogue, a dialogue participant is pursuing a certain goal. The reasons for performing a dialogue act

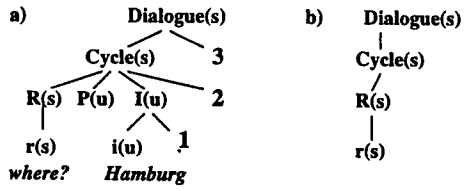

Figure 3: A parse tree and a continuation. Notation: R/r = R/request, P/p = P/promise, I/i = I/inform, *etc.* Parameter values: u = user and s = system. The hearer parameter is left out.

are fairly straightforward in information-seeking dialogues, hence we only find a small set of communicative goals. The basic ones are "provide information" and "seek information", which correspond to Hovy's (Hovy 1988) pragmatic goals `increase knowledge` (of hearer) and `access knowledge` (of hearer). Since we are concerned with the generation of system responses, we ignore user goals for the time being. Further, we focus on the `access knowledge` kind. This goal can be further classified with respect to the kind of information under discussion, i.e. interpreted in the context of the state of the task model. We arrived at the following set of subgoals for the `access knowledge` goal:

1. initiate a choice (by hearer)
2. acquire specific information (from hearer)
3. acquire confirmation (from hearer)
4. acquire disambiguation (by hearer)
5. clarify misrecognition (by hearer)

Obviously, there is no one-to-one mapping between dialogue continuation and communicative goal, since a speaker can use the same continuation to achieve different goals. Continuations merely represent the illocutionary aspect of how a dialogue can continue. They must, however, be interpreted in the context of the current dialogue history to form a concrete communicative goal. Thus, the actual goal depends on the state of the task concept under consideration and the system's beliefs concerning that state. In our previous example, the system employs the same dialogue act (request) to pursue different goals: With the initial request, the system realizes the goal 'acquire specific information' with the instantiation 'of source'. The user supplies an answer (inform), which the system believes to be Homburg. Since its confidence in the result from the speech recognition is low, the continuation is interpreted as 'acquire confirmation' (open end 1), which is instantiated to 'acquire confirmation of recognized source=homburg'.

In the tables below, we summarize how continuations, states of the nodes in task description, and the system's beliefs about the state of the nodes are mapped onto specific communicative goals. If the system continutation ends on a request, we get the following mappings:

State of X	Comm. goals	their domain
open	acquire uval(X)	$\in$ inst(X)
confirm	acquire confirm.	sval(X) = uval(X)
ambiguous	acq. disambig.	sval(X) $\in$ inst(sval(X))
misrecog.	acq. new sval(X)	$\in$ inst(X)\{sval(X)}
closed	acquire sval(Y)	$\in$ inst(Y)

If a continutation ends on an offer, the mappings are:

State of X	Comm. goals	their domain
open	acquire sval(X)	in sub(X)

The following notation applies: inst(X) = instantiations of a concept X; sub(X) = the subconcepts of a concept X; sval(X) = what the system believes the user said; uval(X) = what the user intended, e.g., in the last example above sval(source) = Homburg and uval(source) = Hamburg.

4 Planning an utterance

So far, we have described possible dialogue continuations and interpreted them in the context of the dialogue history as pursuing a particular communicative goal. In most current dialogue systems, each of these goals would be realized as a separate utterance, i.e. the surface structure of the dialogue would merely be a reflection of the underlying dialogue history (see dialogue 1). Our goal, however, is to generate utterances like those in sample dialogue 2. Hence, we need to investigate which communicative goals can be satisfied at a time, in other words, which constellations of dialogue acts given a certain state of the task model can be jointly expressed in one utterance. Recent work on aggregation in the context of natural language generation (e.g.(Dalianis and Hovy 1993)) states that surface structures are abbreviated when information units that in the domain are represented as separate individuals share pertinent features, for instance, syntactical, lexical or semantic features. We extended this notion to allow aggregation of communicative goals: Depending on the common feature, we defined four strategies for condensing dialogue interaction: abbreviation, abstraction, omission, and dominance.

Abbreviation. We call the condensing of information "abbreviation" when a number of continuations that would become adjoining parts of the parse tree and furthermore represent the same communicative goal are expressed in one utterance. If the internal structure of adjoining dialogue cycles (siblings, see Figure 4a) is identical, and the concepts/tasks negotiated in these structures either have the same superconcept or are connected to the same concept by means of a relation, and if the state of the concepts under consideration is open, then the resulting utterance is abbreviated. For instance, abbreviation of several offer constellations that represent the initiate choice as in "Do you want the rate or the total cost of a call?", where 'acquire uval(TESADIS) $\in$ sub(TESADIS)' = 'acquire uval() $\in$ inst(PRICE-PER-MINUTE) $\cup$ inst(TOTAL-COST)' are abbreviated.

Another example would be the abbreviation of acts for acquiring specific information as in "What are the source and the destination of your call?", where 'acquire uval(source)' and 'acquire uval(destination)' are aggregated. Figure 4b, which could be a continuation of the Accept(u) in Figure 4a, illustrates this case.

A third example is abbreviation of several acquire confirmation goals, e.g., "Do you want to call Darmstadt from Magdeburg?"—an abbreviation of 'acquire confirmation sval(source) = uval(source)' 'acquire confir-

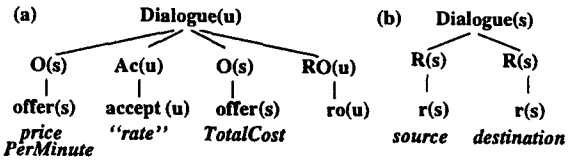

Figure 4: Structures that can be abbreviated.

mation sval(destination) = uval(destination)'.

Abstraction means that we transform several similar goals into a new, more abstract goal. In particular, this applies to the goal 'acquire disambiguation' when a large number of alternatives are at hand. Research in cognitive science and ergonomic design of dialogue systems have shown that human beings can only keep a few alternatives in their short term memory, hence instead of presenting the listener with a long list of alternatives, it is more efficient to phrase a question in a way that avoids mentioning the alternatives. For instance, the goals 'acquire disambiguation sval(name) = maier', 'acquire disambiguation sval(name) = meyer', etc. can be abstracted into 'acquire disambiguation sval(name) in maier,meyer,meier,mayer' with the realization "How do you spell [mai:er]?" instead of "Is [mai:er] spelt with a i, a y, e i, or e y?"

Dominance and subordination. It can be efficient to solicit implicit confirmation of previously recognized values, hence we allow a goal for acquiring a new value and a goal for acquiring confirmation of another value to be realized in one utterance, as in "When do you want to call Frankfurt?" This is a realization of the dominating goal 'acquire uval(startTime)' and the subordinate goal 'acquire confirmation of sval(destination)'.

Omission. Omission means that we leave out a goal altogether, for instance, if the recognition rate of an ambiguous value is high, we take the risk of asking for disambiguation right away, as in the question "Is your call from Frankfurt am Main or Frankfurt an der Oder?" Here the goal 'acquire confirmation sval(source) = uval(source)' is omitted and the goals 'acquire disambiguation of sval(destination) = Frankfurt am Main' 'acquire disambiguation of sval(destination)= Frankfurt an der Oder' have been abstracted as above.

When is aggregation not possible? Above we discussed which structures can be successfully aggregated into more abstract goals and more compact utterances. However, there are certain limits to performing aggregation. For instance, one cannot aggregate between different levels in the dialogue history if the higher level has not yet been satisfied as the following two examples illustrate: *"Do you want the rate or the total cost of a call to where? or ?"When do you want the rate of a call?".

5 Conclusion

In naturally occuring dialogue, the structure of the surface interaction differs from the underlying dialogue history insofar as certain communicative goals are jointly expressed in one utterance, others may even be omitted. We modelled this behaviour for mixed initiative dialogue. In particular, we focused on the system behaviour, and how the system can respond to the user's cooperation, i.e., to newly introduced goals from the user in an equally cooperative manner. We have shown that in order to achieve this behaviour, one has to define constellations of dialogue acts and given a certain state of the task model, which give rise to specific communciative goals. Several of these can be realized within a single utterance.

The next step will be to take a set of communicative goals chosen for aggregation and the content selected by them and pass this to a natural language generation system. For NLG purposes, we will have to investigate how the communicative goals to be realized within one utterance are ranked, how the speech act of an utterance is determined, how the abstraction step alters the content to be expressed, and how different kinds of aggregation rules are realized linguistically. Finally, we will have to build a much more powerful task model in order to support the disambiguation and abstraction procedures, and the generation process.

References

J.F. Allen, B.W. Miller, E.K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proc. of the Annual Meeting of the ACL*, 1996.

M. Blomberg, R. Carlson, K. Elenius, B. Granström, J. Gustafson, S. Hunnicutt, R. Lindell, and L. Neovius. An experimental dialogue system: WAXHOLM. In *Eurospeech'93 (Proc. European Conf. on Speech Communication and Technology)*, pages 1867–1870, 1993.

H. Dalianis and E. Hovy. Aggregation in natural language generation. In *Proc. of the European. Wshp. on Natural Language Generation*, 1993.

E. Hagen. Mixed initiative in a spoken dialogue system. In *Working notes AAAI Spring Symposium Series; Computational Models for Mixed Initiative Interaction* (available as technical report from AAAI), 1997.

E.H. Hovy. *Generating Natural Language under Pragmatic Constraints.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1988.

H. Meng, S. Busayapongschai, J. Glass, D. Goddeau, L. Hetherington, E. Hurley, C. Pao, J. Polifroni, S. Seneff, and V. Zue. WHEELS: A conversational system in the automobile classified domain. In *Proc. of the 1996 Intl. Conf. on Spoken Language Processing (ICSLP'96)*, 1996.

M. Oerder and H. Aust. A realtime prototype of an automatic inquiry system. In *ICSLP'94*, pages 703–706, 1994.

S. Pan and K. McKeown. Spoken language generation in a multimedia system. In *ICSLP'96*, 1996.

M.D. Sadek, A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, and J. Simonin. Effective human-computer cooperative spoken dialogue: AGS demonstrator. In *ICSLP'96*, 1996.

S. Sitter and A. Stein. Modelling the illocutionary aspects of information-seeking dialogues. *Information Processing and Management*, 8(2):165–180, 1992.

S. Whittaker and D. Attwater. Advanced speech applications – the integration of speech technology into complex services. In *Proc. ESCA Wshp. on Spoken Dialogue Systems; Theories and Applications*, pages 113–116, 1994.