

USER MODELLING, DIALOG STRUCTURE, AND  
DIALOG STRATEGY IN HAM-ANS

Katharina Morik  
Technische Universitaet Berlin  
Project group KIT , Sekr. FR 5-8  
Franklinstr. 28/29  
D-1000 Berlin 10 (Fed. Rep. Germany)

ABSTRACT

AI dialog systems are now developing from question-answering systems toward advising systems. This includes:

- structuring dialog
- understanding and generating a wider range of speech acts than simply information request and answer
- user modelling

User modelling in HAM-ANS is closely connected to dialog structure and dialog strategy. In advising the user, the system generates and verbalizes speech acts. The choice of the speech act is guided by the user profile and the dialog strategy of the system.

INTRODUCTION

The HAMBURG Application-oriented Natural language System (HAM-ANS) which has been developed for 3 years is now accomplished. We could perform numerous dialogs with the system thus determining the advantages and shortcomings of our approach (Hoepfner et al. 1984). So now the time has come to show the open problems and what we have learned as did the EUFID group when they accomplished their system (Templeton, Burger 1983). This paper does not evaluate the overall HAM-ANS but is restricted to the aspect of dialog structuring and user modelling.

Dialog structure is represented at two levels: the outline of the dialog is explicitly represented by a top level routine, embedded sub-dialogs are a result of the processing strategy of HAM-ANS. The overall dialog structure is utilized for determining the appropriate degree of detail of the referential knowledge for a particular dialog phase. The embedded sub-dialogs refer to other knowledge sources than the referential knowledge. In the first part of this paper dialog structuring in HAM-ANS is described. Handling of dialog phenomena as ellipsis and anaphora is not

discussed here, but see (Jameson, Wahlster 1982). The second part of this paper presents user modelling with respect to a dialog strategy which selects and verbalizes the appropriate speech act of recommendation.

DIALOG STRUCTURE

In one of its three applications HAM-ANS plays the role of a hotel clerk who advises the user in selecting a suitable room. The task of advising can be seen here as a comparison of the demands made on an object by the client and the advisor's knowledge about available objects, performed in order to determine the suitability of an object for the client. Dialogs between hotel clerks at the reception and the becoming hotel guest are usually short and stereotyped but offer some flexibility as well because the guests do not establish a homogeneous group. With recourse to this restricted dialog type we modelled the outline of the dialog. Dialog structure is not represented in terms of some actions the user might want to perform as did Grosz (1977), Allen (1979), Litman, Allen (1984) nor in terms of information goals of the user as did Pollack (1984), but we represent and use knowledge about a dialog type. For formal dialogs in a well defined communication setting this is possible. For a practical application the dialog phases and steps should be empirically determined. We do not consider the hotel reservation situation an example for real application. We just wanted to show the feasibility of recurring to linguistic knowledge about types of texts or dialogs. Real clerk - guest dialogs show some features we did not concern. Features of informal man-man-communication as, e.g., narratives and role-defining utterances of dialog partners were excluded from the model of the dialog. Man-machine-interaction is seen as formal as opposed to informal communication, and there is no way of redefining it as personal talk.

The outline of the dialog is a structure at three different levels: there are three dialog phases, each consisting of several dialog steps (see Fig.1). Each dialog step can be performed by several dialog acts.

Although the outline of the dialog is fixed, there is also flexibility to some extent:

\* The work on HAM-ANS has been supported by the BMFT (Bundesministerium fuer Forschung und Technologie) under contract 08it15038.

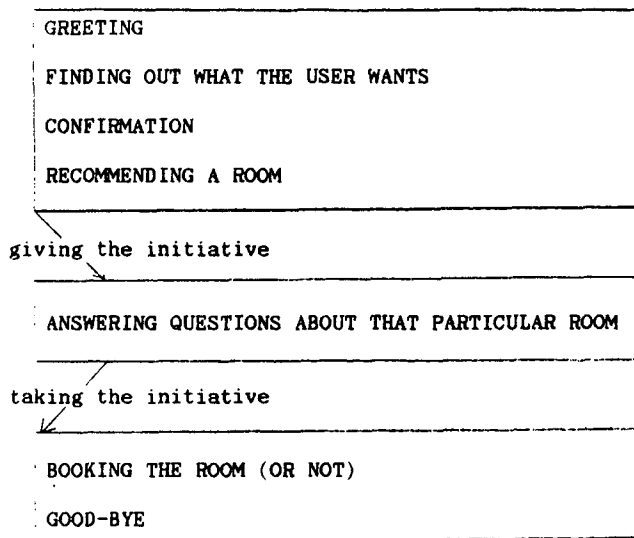


Fig.1 outline of the dialog

- The dialog step "Finding out what the user wants" consists of as many questions of the system as are necessary
- If the confirmation step does not succeed it is jumped back to the dialog act where the user initializes the dialog step "Finding out what the user wants"
- In the dialog phase concerning a particular room the systems asks for regaining the initiative. If the user denies the questioning phase is continued.

The advantages of fully utilizing knowledge about the dialog type are the reduction of complexity, i.e. the system does not have to recognize the dialog step, realistic response time because no additional processing has to be done for planning the dialog, and the explicit representation of the dialog structure. The declarative representation of dialog structure allows for modelling different degrees of detail of the world knowledge attached to the dialog phases.

#### Views of the domain

We believe that the degree of detail of referential knowledge is constituting a dialog phase. In other words, different degrees of detail or abstraction separate dialog phases. Reichman could have made this point, because her empirical data do support this observation (Reichman 1978). In focus is not only a certain portion of a task tree or a certain step in a plan, but also a certain view of the matter. Therefore, attached to the dialog phases are different knowledge bases accessible in these phases. World knowledge contains for the first and the last phase overview knowledge about the hotel and its room categories, for the second phase detailed knowledge about one instance of a room category, i.e. a particular room.

The room categories are derived from the individual rooms by an extraction process which disregards location and special features of objects as, e.g., colour. But representing overview knowledge is not just leaving out some types of arcs in the referential semantic network! One capability of the extraction process is to group objects together if there is an available word to identify the group and to identify objects which are members of the group not just parts of a whole. An example may clarify this.

One advantage of some of the rooms is that they have a comfortable seating arrangement made up of various objects: couch, chairs, coffee table, etc. HAM-ANS can abstract from this grouping of objects and identify it as a "Sitzecke" - a kind of cozy corner, a common concept and an every-day word in German. Another example of a group is the concept "Zimmerbar" (room bar) consisting of a refrigerator, drinking glasses and drinks.

Another difference between overview knowledge and detailed knowledge is that some properties of objects are inherited to the room category. For example, what can be seen out of the window is abstracted to the view of the room category. A room category has a view of the Alster, one of the two rivers of Hamburg, if at least one window faces the Alster.

While selecting a suitable room for the user the system accesses the abstracted referential knowledge. Not until the dialog focuses on one particular room, does the system answer in detail questions about, e.g. the location of furniture, its appearance, comfort, etc. Thus different degrees of detail are associated with different dialog phases because the tasks for which the referential knowledge is needed differ. The link between the overview information, e.g. that a room category has a desk, a seating arrangement ("Sitzecke") etc., and the detailed referential knowledge about a particular room of that category, e.g., that there is a desk named DESK1, that there are three arm chairs and a coffee table etc., is established by an inverse process to the extraction process. This inverse process finds tokens or derives implicit types, for which in turn the corresponding tokens are found. When initiative is given to the user, the tokens of objects mentioned in the preceding dialog are entered into the dialog memories which keep track of what is mutually known by system and user. Thus, if the seating arrangement ("Sitzecke") has been introduced into the dialog the user may ask, where "the coffee table" is located using the definite description, because by naming the seating arrangement the coffee table is implicitly introduced.

The procedural connection between overview and detailed knowledge entails, however, a problem. First, while semantic relations between concepts are represented in the conceptual network thus determining noun meaning, the meaning of "group nouns" could not be represented in the same formalism. Second, inverting the extraction process and entering tokens into a dialog memory

leads to a problem of ambiguous referents. If a "Sitzecke" has been mentioned - which arm chairs or couches are introduced and how many? The system may infer the tokens, but not the user. For him, a default description of a "Sitzecke", which is concretized only if an object is named by the user, should be entered into the dialog memory.

#### Subdialogs

We have seen the outline of the dialog, but also inside the questioning phase there is a dialog structure. The system initiates a clarification dialog if it could not understand the user input. This could be, for instance, a lexicon update dialog. The user may start a subdialog in putting a meta-question as, e.g., "What was my last question?" or "What is meant by carpet?". Meta-questions are recognized by clue patterns. Here, too, attached to the subdialogs are different knowledge sources: subdialogs are not referring to the referential knowledge (about a particular room) but to the lexical update package, the dialog memories, or the conceptual knowledge.

Subdialogs are embedded dialogs which can be seen in the system behavior regarding anaphora, for instance. They are processed in bypassing the normal procedure of parsing, interpretation and generating. This solution should be replaced by a dialog manager module which decides as a result of the interpretation process which knowledge source is to be taken as a basis for finding the answer.

### USER MODELLING

User modelling in AI most often concerns the user's familiarity with a computer system (Finin 1983, Wilensky 1984) or his/her knowledge of the domain (Goldstein 1982, Clancey 1982, Paris 1983). These are, of course, important aspects of user modelling, but the system must in addition model the user-assessment aspect.

#### Value judgements

The claim of philosophers and linguists (Hare 1952, Grewendorf 1978, Zillig 1982) that value judgements refer to some sort of an evaluation standard are not sufficient. In AI, the questions are:

- how to recognize evaluation standards
- how to represent them
- how to use them for generating speech acts

It is well known that recommendations rely on the presumed evaluation standards of the hearer. If these speech acts are to be actually implemented in the system and not just read into the system responses by the user, the system must be able to recognize the user's standards of evaluation, derive the demands placed by the user on objects or situations which form the subject with the demands. It would be overinformative to output evaluations about every possible object. Rather,

evaluations should be used to select those objects which might interest the user. For example, which information about a hotel room is presented to the user depends on the interests of the user and his requirements, which can be inferred from his/her evaluation standard. The information to be outputted can be selected on the basis of user's requirements rather than on the basis of freedom of redundancy given the user's knowledge. Thus, the choice of the relevant objects as well as the choice of the appropriate value judgement requires the modelling of the user's evaluation standards.

A system which performs recommendations of fiction books is Rich's GRUNDY (Rich 1979). The basis heuristic underlying the consultative function is that people are interested in books in which characters have the same type of personality as they themselves have, or where characters are facing a situation similar to their own. Therefore, recognizing the personality type of a user is a central concern for GRUNDY and can be used directly for the evaluation of books. We'll see that for HAM-ANS the utilization of knowledge about the user is not so straightforward. Neither is HAM-ANS interested in the personality type of the user nor is there any plausible direct match between personality type and room category. We want to distinguish the user facts, which is knowledge about the user and his wants given explicitly by himself, the user profile, which is knowledge about the user inferred by the system, and the interest profile, which is the user's evaluation standard as it is inferred by the system. One has to be inferred from the other.

#### Building the interest profile

Let us look at an example of the first dialog phase:

```
SYS : HOTEL ALSTERBLICK, GUTEN ABEND!  
      At the sign of Hotel Alsterblick, good  
      evening!  
USER: Hallo hier Doktor Heine von IBM Heidelberg  
      Hello this is Doctor Heine from IBM  
      Heidelberg  
SYS : WAS KANN ICH FUER SIE TUN?  
      What can I do for you?  
USER: Ich moechte ein Doppelzimmer mit Bad fuer  
      zwei Tage.  
      I would like a single room with bath for two  
      days.  
SYS : UND AB WANN WIRD DAS ZIMMER BENOETIGT?  
      And for when would you like the room?  
USER: Ab morgen  
      For tomorrow  
SYS : UND AUF WELCHEN NAMEN SOLL DAS ZIMMER  
      GEBUCHT WERDEN?  
      And in whose name should the room be booked?  
USER: Auf meinen.  
      In mine.  
SYS : EIN DOPPELZIMMER BIS ZUM 25.4. HAETTEN SIE  
      GERN?  
      You would like a double room until april 25?  
USER: Ja.  
      Yes.  
SYS : UNSER HOTEL WIRD IHNEN GEFALLEN UND WIR
```

HABEN AUCH NOCH EIN ZIMMER FREI. DAS ZIMMER HAT KEIN FERNSEHGERAET ABER EINE ZIMMERBAR, EINEN SCHREIBTISCH, EIN TELEFON UND EINE SITZECCKE. HABEN SIE ZU DIESEM ZIMMER NOCH EINE FRAGE?

You will enjoy our hotel and there is still a room available. The room has no TV but a room bar, a desk, a telephone, and a seating arrangement. Do you have questions concerning this room?

In this dialog phase the user first informs the system about his concerns and indirectly give some hints about his standards. The facts, i.e. from when to when what kind of a room should be booked for whom, are stored in a data structure under the heading GUEST. If the user has not given all the necessary information, the system asks for it.

The way the user introduces himself may give a hint as to what kind of user he is. But the system would not ask for title or firm or location if the user has not volunteered the information. From these data some inference processes are initiated, estimating independently profession (here, manager) financial status (here, rich), and purpose of trip (here, transit). The estimations are stored under the heading SUPPOSE. They can be viewed as stereotypes in that they are characteristics of a person, relating him/her to a certain group to which a number of features are assigned (Gerard, Jones 1967). As I mentioned earlier the application of stereotypes is not as straightforward as in Rich's approach. Two steps are required. We've just seen the first step, the generation of SUPPOSE data. As opposed to GUEST data, the SUPPOSE data are not certain, thus supposed data and facts are divided.

In the second step, each of the SUPPOSE data independently triggers inferences, that derive requirements presumably placed on a room and on the hotel by the user. The requirements are roughly weighed as very important, important and surplus (extras). If the same requirement is derived by more than one inference and with different weights, the next higher weight is created or the stronger weight is chosen, respectively. This is, of course, a rather simplified way of handling reinforcement. But a more finely treatment would yield no practical results in this domain. The requirements for the room category and for the hotel are stored separately in semantic networks. An excerpt of the networks corresponding to the dialog example above:

```
((WICHTIG (HAT Z FERNSEHGERAET).1)
((SEHR-WICHTIG (HAT Z TELEFON).1)
((SEHR-WICHTIG (HAT Z SCHREIBTISCH).1)

((SURPLUS (HAP HOTEL1 FREIZEITANGEBOT-2).1)
((SEHR-WICHTIG (IST HOTEL1 IN/ ZENTRALER/ LAGE).1)
```

The requirements are then tested against the knowledge about the hotel and the room categories. Some requirements correspond directly to stored features. Others as here, for instance, the leisure opportunities number 2 or the central

location of the hotel are expanded to some features by inference procedures. Thus here, too, there is an abstraction process. The requirements together with their expansions represent the concretized evaluation standard of the user. They are called the interest profile of the user.

#### Generating recommendations

Now, let's see what the system does with this. First, it matches the requirements against the room or hotel features thus yielding an evaluation from every room category of the requested kind (here, double room). The evaluation of a room category consists of two lists, the one of the fulfilled criteria and the one of the unfulfilled criteria.

Secondly, based on this evaluation speech acts are selected. The speech act recommendation is split up into STRONG RECOMMENDATION, WEAK RECOMMENDATION, RESTRICTED RECOMMENDATION, and NEGATIVE RECOMMENDATION. The speech acts as they are known in linguistics are not fine grained enough. Having only one speech act for recommending would leave important information to the propositional content. The appropriate recommendation is chosen according to the following algorithm:

- if all the criteria are fulfilled, a STRONG RECOMMENDATION is generated
- if no criteria are fulfilled, a NEGATIVE RECOMMENDATION is generated
- if all very important criteria are fulfilled, but there are violated criteria, too, a RESTRICTED RECOMMENDATION is generated
- if there are some criteria fulfilled, but even very important criteria are violated, a WEAK RECOMMENDATION is generated

This process is executed both for the possible room categories and for the hotel. The result is a possible recommendation for each room category and the hotel. Out of these possible recommendations the best choice is taken.

Third, a rudimentary dialog strategy selects the most adequate speech act for verbalization. For instance, if there is nothing particularly good to say about a room but there are features of the hotel to be worth landing, then the hotel will be recommended. The hotel recommendation is only verbalized, iff it suits perfectly and the best possible recommendation for a room category is not extreme, i.e. neither strong nor negative. The negative recommendation has priority over the hotel recommendation because an application-oriented system should not persuade a user nor has a goal for its own - although this is an interesting matter for experimental work and can be modelled within our framework.

In our example dialog the best recommendation of a room category is the restricted recommendation for room category 4. The hotel fulfills all the inferred requirements and can be recommended strongly. These speech acts have to be verbalized now. For verbalization, too, a dialog strategy is

applied:

The better the evaluation  
the shorter the recommendation.

The most positive recommendation is realized as:

DA HABEN WIR GENAU DAS PASSENDE ZIMMER FUER SIE  
FREI.  
(We have just the room for you.)

In our example the recommendation can't be so short, because the disadvantages should be presented to the user so that he can decide whether the room is sufficient for his requirements. Therefore, the room category is described. Among all the features of the room only those are verbalized which correspond to the user's interest profile. In order to verbalize the restricted recommendation, a "but" construct of the internal representation language SURF of HAM-ANS is built (Fig.2).

From this structure the HAM-ANS verbalization component creates a verbalized structure which is then transformed into a preterminal string from which the natural language surface is built and then outputted (Busemann 1984). The verbalization component includes the updating of the dialog memories.

```
(af-d: IS
(t-s: (q-d: D-) (lambda: x4 (af-a: ISA x4
ZIMMER)))
(lambda: x4
(af-a: HAT x4
(t-o: BUT
(t-s: (q-qt: KEIN) (lambda: x4 (af-a: ISA x4
FERNSEHGERAET)))
(t-o: AND
(t-s: (q-qt: E-) (lambda: x4 (af-a: ISA x4
ZIMMERBAR)))
(t-o: AND
(t-s: (q-qt: E-) (lambda: x4 (af-a: ISA x4
SCHREIBTISCH)))
(t-o: AND
(t-s: (q-qt: E-) (lambda: x4 (af-a: ISA x4
TELEFON)))
(t-s: (q-qt: E-) (lambda: x4 (af-a: ISA x4
SITZECKE))))))))))
```

Fig.2 SURF structure

After the dialog strategy has selected the most positive recommendation it can fairly give regarding the evaluation form of the room category which suits best the (implicit) demands of the user and has chosen the appropriate formulation for the recommendation, it prepares to give the initiative to the user thus entering the questioning dialog phase. That is: it is focused on the selected room category and the more detailed data about an instance of that particular room category are loaded. In our example, the referential network and the spatial data of room 4 are accessed.

## OPEN QUESTIONS

The problems that have yet to be solved may be divided into three groups: those that could be solved within this framework, those that require a change in system architecture and those that are of principle nature.

A problem which seems to fit into the first group is the explanation of the suppositions. The user should get an answer to the questions:

Who do you think I am?  
How do you come to believe that I am a manager?  
How do you come to believe that I need a desk?

The first question may be answered by verbalizing the SUPPOSE data. The second and the third question must be answered on the basis of the inferences taking into account the reinforcement as did Wahlster (1981).

The third question may be a rejection of the supposed requirement rather than a request for justification or explanation. Understanding rejections of supposed requirements includes the modification of the requirement networks. For example, the user could say after the restricted recommendation:

But I don't need a TV!

Then the room category 4 fits perfectly well and may be strongly recommended.

Or the user could state:

But I don't want a desk. I would like to have a TV instead.

In this case the requirement net of the room categories is to be changed:

```
REMOVE ( ? (HAT Z DESK))
ADD (VERY-IMPORTANT (HAT Z TV))
```

With this the room categories have to be evaluated again and perhaps another room category will then be recommended.

A type of requirements that is yet to be modelled is the requirement that something is not the case. For example, the requirement that there should be no air-conditioning (because it's noisy).

A change in system architecture is required if the advising is to be integrated into the questioning phase. The reason why this is not possible by now is, for one part, of practical nature: memory capacity does not allow to hold overview knowledge and detail knowledge at once. The other part, however, is the increase of complexity. Questions are then to be understood as implicit requirements. For example:

The room isn't dark?

```
ADD ( IMPORTANT ( IST Z BRIGHT))
```

The hard problem we are then confronted with is an

instance of the frame problem (Hayes 1971:495, Raphael 1971). When does the overall evaluation of a room category not hold any longer? When are all the room categories to be evaluated again according to a modified interest profile? When should it be switched from one selected room category to another? These are problems of principle nature which have yet to be solved. Further research is urgently needed.

#### REFERENCES

- ALLEN, J.F.(1979): A Plan Based Approach to Speech Act Recognition. Univ. of Toronto, Techn.Rep.No.131/79.
- BUSEMANN, S.(1984): Surface Transformations During The Generation Of Written German Sentences. Hamburg Univ., Research Unit for Information Science and Artificial Intelligence, Rep. ANS-27.
- CLANCEY, W.J. (1982): Tutoring Rules For Guiding A Case Method Dialogue. In: D. Sleeman, J.S. Brown (eds): Intelligent Tutoring Systems, pp.201
- FININ, T. (1983): Providing Help and Advice in Task Oriented Systems. In: Procs. 8th IJCAI, Karlsruhe, pp.176.
- GERARD, H.B., JONES, E.E. (1967): Foundations of Social Psychology. New York, London.
- GOLDSTEIN, I. (1982): The Genetic Graph: A Representation For The Evaluation Of Procedural Knowledge. In: D. Sleeman, J.S. Brown (eds) Intelligent Tutoring Systems, pp.51.
- GREWENDORF, G. (1978): Zur Semantik von Wertausserungen. In: Germanistische Linguistik 2-5, pp.155.
- GROSZ, B.J. (1977): The Representation And Use Of Focus In Dialog Understanding. SRI, Techn. Note No. 151.
- HARE, R.M. (1952): The Language Of Morals. German (1972), Frankfurt a.M.
- HAYES, P. (1971): A Logic Of Actions. In: B. Meltzer, D. Michie (eds): Machine Intelligence 6, pp.495.
- HOEPPNER, W., CHRISTALLER, T., MARBURGER, H., MORIK, K., NEBEL, B., O'LEARY, M., WAHLSTER, W. (1983): Beyond Domain-Independence: Experience With The Development Of A German Language Access System To Highly Diverse Background Systems. In: Procs. 8th IJCAI, Karlsruhe, pp. 588.
- JAMESON, A., WAHLSTER, W. (1982): User Modelling In Anaphora Generation: Ellipsis And Definite Description. In: Procs. ECAI-82, Orsay, pp.222.
- LITMAN, D.J., ALLEN, J.F. (1984): A Plan Recognition Model For Clarification Subdialogs. In: Procs. COLING-84, Stanford, pp. 302.
- PARIS, J.J. (1983): Determining The Level Of Expertise For Question Answering. New York (no report number).
- POLLACK, M. E. (1984): Good Answers To Bad Questions: Goal Inference In Expert Advice-Giving. In: Procs. Canadian Conference on AI, pp.20.
- RAPHAEL, B. (1971): The Frame Problem in Problem Solving Systems. In: N. Findler, B. Meltzer (eds): Artificial Intelligence and Heuristic Programming, pp. 159.
- REICHMAN, R. (1978): Conversational Coherency. In: Cognitive Science 2, pp.283.
- RICH, E. (1979): Building And Exploiting User Models. Carnegie Mellon Univ. Rep. No. CMU-CS-79-119.
- TEMPLETON, M., BURGER, J. (1983): Problems In Natural-Language Interface To DBMS With Examples From EUFID. In: Procs. Conference on Applied Natural Language Processing, Santa Monica, pp.3.
- WAHLSTER, W. (1981): Natuerlichsprachliche Argumentation in Dialogsystemen - KI-Verfahren zur Rekonstruktion und Erklaerung approximativer Inferenzprozesse. Berlin, Heidelberg, New York.
- WILENSKY, R. (1984): Talking To UNIX In English: An Overview Of An Online UNIX Consultant. In: The AI Magazine, Vol.V, No. 1, pp.29.
- ZILLIG, W. (1982): Bewerten - Sprechakttypen der bewertenden Rede. Tuebingen.