

Cooperative Access Systems

W. WAHLSTER

Universitaet des Saarlandes, Computer Science Department
Im Stadtwald 15, D-6600, Saarbruecken 11
Federal Republic of Germany

In the next five years we can expect to find a variety of cooperative natural language (NL) interfaces to database systems, expert systems, operating systems, CAD systems and even text formatting systems. Although the general problem of getting computers to understand NL is far from being solved, the technology for limited NL access systems is available now and there is a large growing market for such cooperative interfaces to a wide variety of software. In the first part of the paper, we illustrate the capabilities for cooperative response generation implemented in AI systems. Then we briefly review the state of the art in NL interfaces to databases and expert systems. Topics covered include user modeling, the generation of explanations, mixed initiative dialogs and knowledge acquisition via NL communication.

1. Introduction

Cooperative interfaces are developed with a view to increasing the accessibility of valuable computer facilities, such as database systems [9], CAD systems [3] and even text formatting systems. A further application area is that of natural language interfaces to expert systems. An example here is the system XCALIBUR [2], an interface to the configuring system XCON. Yet another area involves cooperative communication with operating systems, such as VMS or UNIX. Examples are the AI system UC (UNIX Consultant [19]) and the system Wizard [6] for VMS. What knowledge and capabilities does a system require if it is to exhibit cooperative behavior? First, it requires relevant conceptual and inferential knowledge: It must have a model of the person with whom it is to cooperate, and it must be able to understand the goals and the actions of this person and to adapt its own behavior accordingly. Research on such interfaces concentrates on the one hand on problems of man-machine interaction; on the other hand, questions arise which are typical of artificial intelligence research, e.g., knowledge representation, knowledge acquisition and knowledge processing, as well as knowledge presentation.

2. The Generation of Cooperative Responses

I will now try to illustrate what is meant by cooperative behavior in an interface by presenting several examples of such behavior which have been realized in a number of AI systems in various parts of the world (see Fig. 1, cf. also [1, 18]).

Let us begin with an example in which the system calls the user's attention to the violation of a presupposition. If the user asks the interface a question like (1) (see Fig. 1), and the system replies with (2), we can speak of cooperative behavior. If you pose the same question to one of today's conventional information systems, you will probably simply receive the answer 'No'. This answer, however, is misleading, since

an implicit assumption, or presupposition, on which the question is based is not fulfilled - namely that an invoice has been sent out to the client in the first place. In other words, the response of a traditional information system would give the user a completely false impression. A cooperative system must be in a position to call the user's attention to this sort of violation of a presupposition.

<p>Pointing out incorrect presuppositions</p> <p>(1) U: Did Mr. Jones pay the bill for our last delivery?</p> <p>(2) C: No, we have not yet sent Mr. Jones the bill.</p> <p>(3) S: No. (→ misleading conclusion)</p> <p>Indication of discontinuities in the database</p> <p>(4) U: Is there a seat available on the flight to New York on August 8th?</p> <p>(5) C: Yes, but from August 9th onwards, tickets are \$200 cheaper due to the winter season.</p> <p>(6) S: Yes. (→ unfriendly response)</p> <p>Unsolicited justification of an answer</p> <p>(7) U: Is a long-term fixed money investment worthwhile at the moment?</p> <p>(8) C: No, because interests are tending to rise.</p> <p>(9) S: No. (→ missing argumentation)</p> <p>Additional information on the individual conjuncts</p> <p>(10) U: Are Mr. Brown and Mr. Gray on a business trip?</p> <p>(11) C: No, Mr. Brown is on a business trip but Mr. Gray is here.</p> <p>(12) S: No. (→ answer provoking follow-up questions)</p> <p>Unsolicited filling of the relevant deep case</p> <p>(13) U: Which colleagues are on business trips?</p> <p>(14) C: Mr Baker is in London, Mr. Miller in New York and Mr. Smith is in Japan.</p> <p>(15) S: Mr. Baker, Mr. Miller and Mr. Smith. (→ answer provoking follow-up questions)</p> <p>Offering to 'monitor' for information requested</p> <p>(16) U: Is a seat available on the last flight from Munich to Hamburg?</p> <p>(17) C: No, not at the moment, but I will put you on the waiting list and I will let you know if a passenger cancels his flight.</p> <p>(18) S: No. (→ unfriendly response)</p> <p>Using more specific quantifiers</p> <p>(19) U: Are you employing some researchers with a Ph. D., too?</p> <p>(20) C: Yes, three.</p> <p>(21) S: Yes. (→ less cooperative response)</p>
--

Fig. 1. Forms of Over-Answering Implemented in AI Systems.
(U = user, S = conventional system, C = Cooperative System).

Another way in which a system can be cooperative is by pointing out a discontinuity in the database. Suppose, for example, you ask the system a question like (4), it would then be cooperative for the system to respond with (5).

Siklossy developed such a system several years ago: It searches the database systematically for such discontinuities, and when it determines that a small change might have substantial positive consequences for the user, it points this fact out, thereby over-answering the question [12].

We have investigated and implemented several further forms of cooperative behavior in our own project. One of these involves the explanation of an answer. A system should spontaneously explain its own answers when such an explanation is likely to be useful to the user [13]. For example, if you ask an expert system for investment consulting a question like (7), it should not simply answer 'No', it should go on to add a justification like (8).

The part of a system which generates elucidations of this general sort is called an explanation component [14].

Still another strategy is to supply additional information about individual conjuncts within a question. If you ask a traditional information system a question like (10), the answer will be simply 'No'.

But one would generally like to know why this conjunction of statements is not true. It is therefore more helpful if the system replies with (11). That is, the system should make it clear which of the conjuncts is responsible for the falsity of the conjunction [15]. A further possible strategy is to fill in deep cases in the semantic representation of a question. The purpose of this is to anticipate obvious follow-up questions by over-answering the original question. If the user asks, for example, a question like (13), the system can in principle interpret the question literally and supply just the names of these staff members (see (15) in Fig. 1). But it is in general more cooperative, given a question about a trip, to supply the destination of the trip as well (see (14), [15]).

Yet another form of cooperative behavior has been implemented by B. Webber and colleagues at the University of Pennsylvania [10]. It takes the form of an offer by the system to keep track of the future status of a condition specified by the user. If you ask question like (16), the system should reply with (17). This process is termed the monitoring of a condition: The system becomes active without further prompting as soon as the condition specified by the user is fulfilled.

A final possibility is the replacement of vague quantifiers which occur in the question by more precise ones. Suppose you ask a question like (19), the quantifier 'some' can then be made more precise the system over-answers with (20). This answer contains much more information of course, than simple 'Yes' [15].

3. Mixed-Initiative Natural Language Dialog Systems

Let us now look at the overall structure of systems of this sort (see Fig. 2). Such a system has, first, an analysis component which translates the natural language input into a formal semantic representation. This process is knowledge-based; the knowledge base contains general background knowledge, including a semantically

oriented grammar and, e.g. a semantic network which encodes the relationships among the system's concepts. There is also knowledge which applies only to a particular domain of discourse; this includes domain-specific inference rules and the system's referential knowledge. Finally, part of the knowledge base contains so-called dialog knowledge. This includes a representation of the momentary thematic focus in the dialog, a user model which contains information about the user [17], and finally an inference memory which enables the system to explain the reasoning underlying its answers.

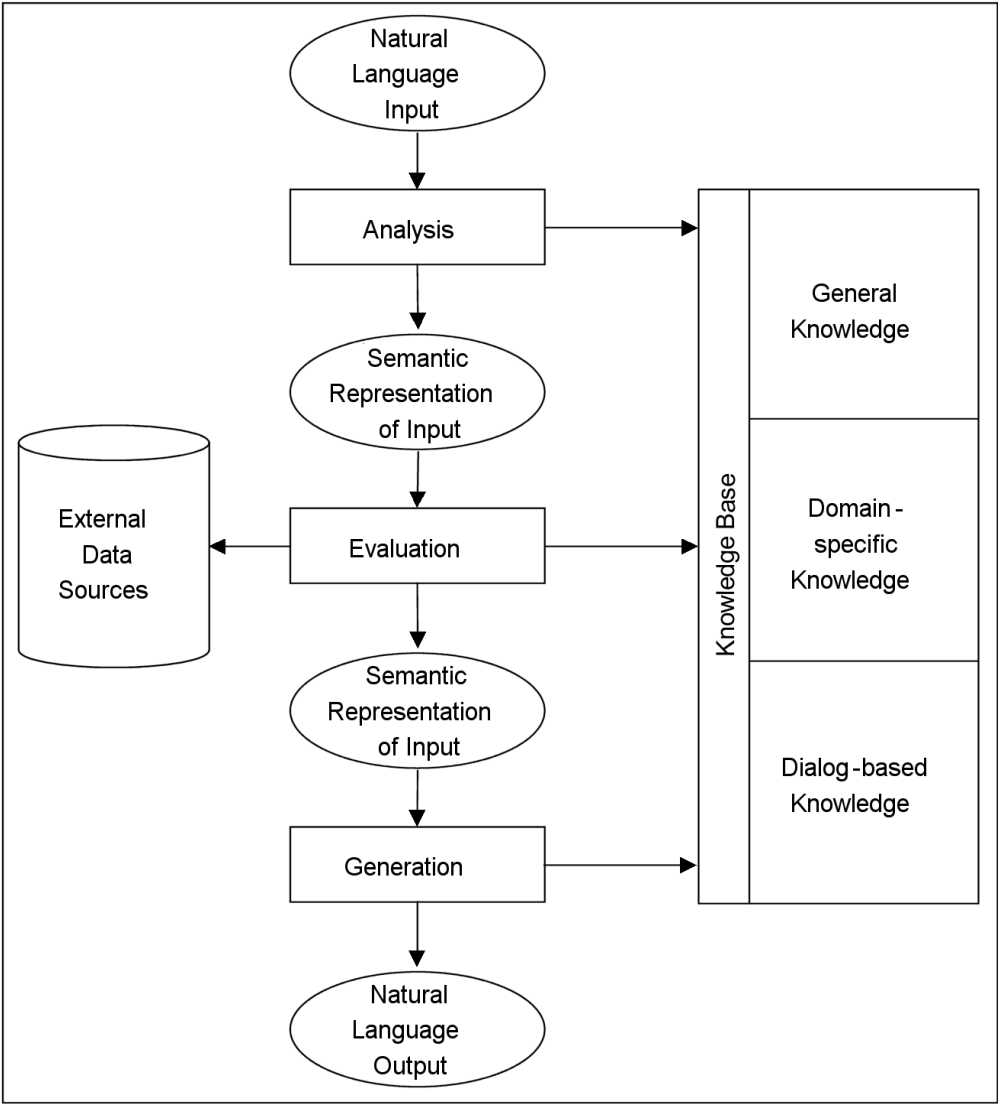


Fig. 2. The Overall Structure of Natural Language Dialog Systems.

The output of the analysis component is passed to an evaluation component. This component takes as input the semantic representation of the question and returns the semantic representation of the answer. It accesses not only the knowledge base sketched above but also external sources of data (e.g., a database) which belong to the back end system that the user is primarily interested in accessing. Finally, a generation component is required which translates the formal semantic representation of the answer into a natural language utterance suitable for output.

As an example of such a system, I would like to discuss briefly our own system HAM-ANS* [7], with which it is possible to conduct written dialogs about several different domains of discourse (see Fig. 3). We have chosen both domains involving mainly everyday knowledge and domains which require specialized knowledge. As for the type of user who is given access to the system, we have chosen both laymen and experts. The system can be run in a mode in which its own interests are paramount in determining the form of its answers or in a mode in which its behavior is purely cooperative. We have used the following dialog situations as case studies: The user can reserve a room in a hypothetical hotel; ask questions about a traffic scene which is being monitored by a vision system that analyzes image sequences; or query a relational database in which mass data about fishing expeditions are stored. An important point is that the system actually includes a switch, i.e. we can actually move from one application to another. For this to be possible, the structure of the system has to be so modular that only certain parts of the knowledge base need to be replaced to adapt the interface to an entirely different back end system.

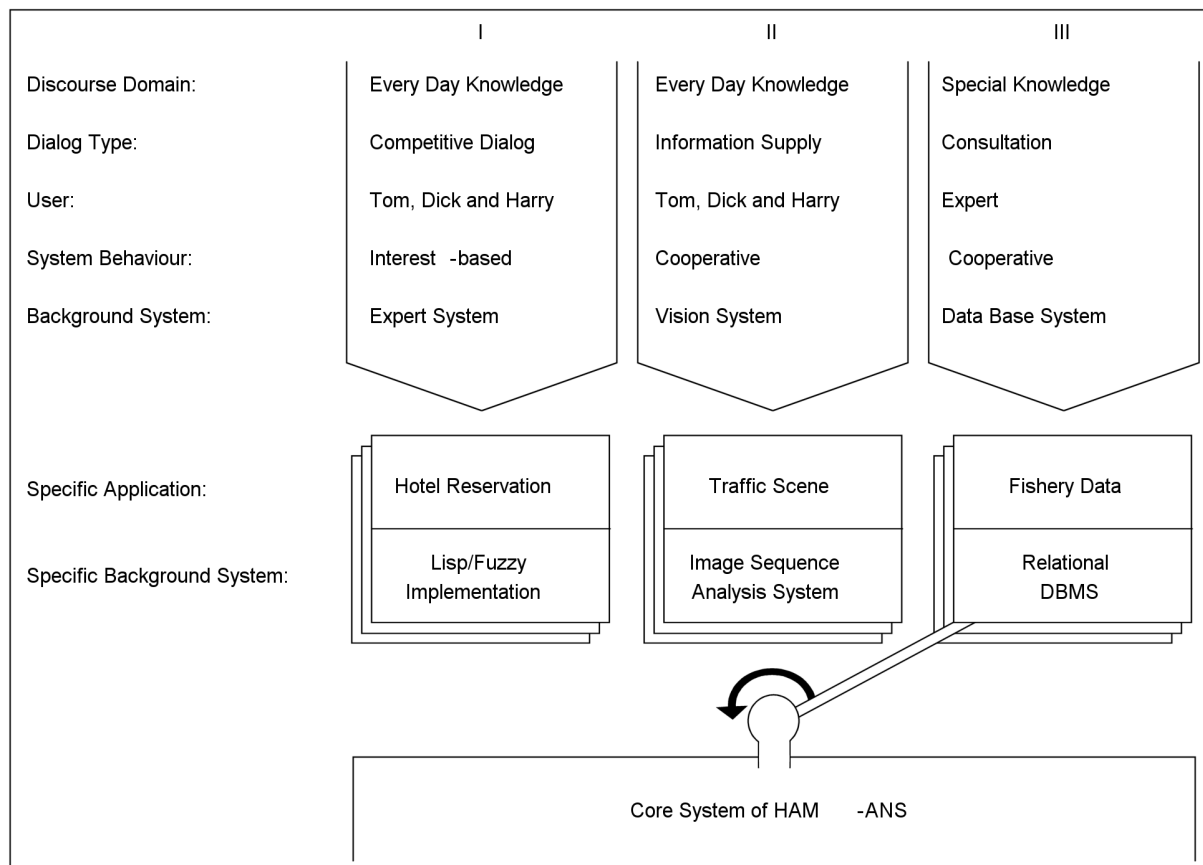


Fig. 3. The Discourse Domains of HAM-ANS.

One goal of our research is to have the system engage in mixed-initiative dialogs. That is, neither the system nor the user should retain the initiative all the time, asking questions to be answered by the other. Instead, in accordance with the turn-taking rules which govern dialogs between human speakers, the system should under certain circumstances yield the initiative to the user and then seize it again later. A clear case where this is necessary is the hotel reservation situation, where first the

* Research on HAM-ANS is currently being supported by the German Ministry of Research and Technology (BMFT) under contract 081T15038.

user calls a particular hotel and then the system takes the initiative by asking certain questions to determine which room seems most suitable. The user then has the opportunity to ask questions in order to evaluate the room which has been suggested. Finally, the system will try to regain the initiative by asking whether the user wishes to reserve the room or not. Some of the challenging problems that the system has to deal with in the course of such a dialog involve the general topic of user modelling, which I mentioned above. Suppose for example, that the user asks a question like

(22) U: What kinds of chairs does the room have?

(23) C: There are several comfortable chairs in the room.

In order to generate an appropriate description of the chairs, the system must make some assumptions about the user's intended activities in the room [8]. For example, if information has accumulated in the user model which suggests that the user might intend to entertain guests, it is reasonable to suppose that the user is particularly interested in knowing how comfortable the chairs are. The system should answer in such a case with (23).

As you can see, the system's behavior here is interest-based, i.e. guided by the system's own interest in renting the room.

The necessity of taking into account the user's desires becomes even clearer when we look at questions by the user which contain expressions that reflect certain expectation. If, for example, the user, phoning a luxury hotel, asks

(24) U: Does the room by any chance have a TV?

(25) C: Yes, of course.

(26) S: Yes.

the system is justified in drawing certain conclusions on the basis of the occurrence of the expression 'by any chance'. First, the user apparently would like the room to have a TV. Second, the user evidently unaware that he or she is talking to a luxury hotel since in such a hotel there is normally a TV in every room, so that it would be inappropriate to formulate the question so cautiously. In such a case, therefore, the system should not reply with (26), but rather with (25). This more emphatic answer has the function of indicating that the user's uncertainty was based on a incorrect expectation, that the question in principle needn't have been asked at all.

4. Intelligent Interfaces to Expert Systems and Databases

An important application of cooperative natural language interfaces is to provide access to expert systems. The intelligent interface provides access to the inference component, the explanation component, and the knowledge-acquisition component (see Fig. 4). The dialog component also makes it possible for both the user and the system designer to access the contents of the knowledge base directly. We have just completed the preparations for a system called XTRA (EXpert TRANslator, cf. [16]) which is designed to give both the user and the expert access to a variety of expert

systems and to be so adaptable that it can also be used to control a general expert system shell (see Fig. 5).

As you see in Fig. 4, we have two separate knowledge bases, one for the natural language interface and a second for the expert system. This separation of the knowledge bases makes it relatively easy to adapt the system for a new task: Only the task-specific knowledge needs to be replaced. The primary communication medium is natural language, typed in and printed out at a computer terminal. In addition, input may be supplied with the help of a mouse, and output may make use of high-resolution graphics.

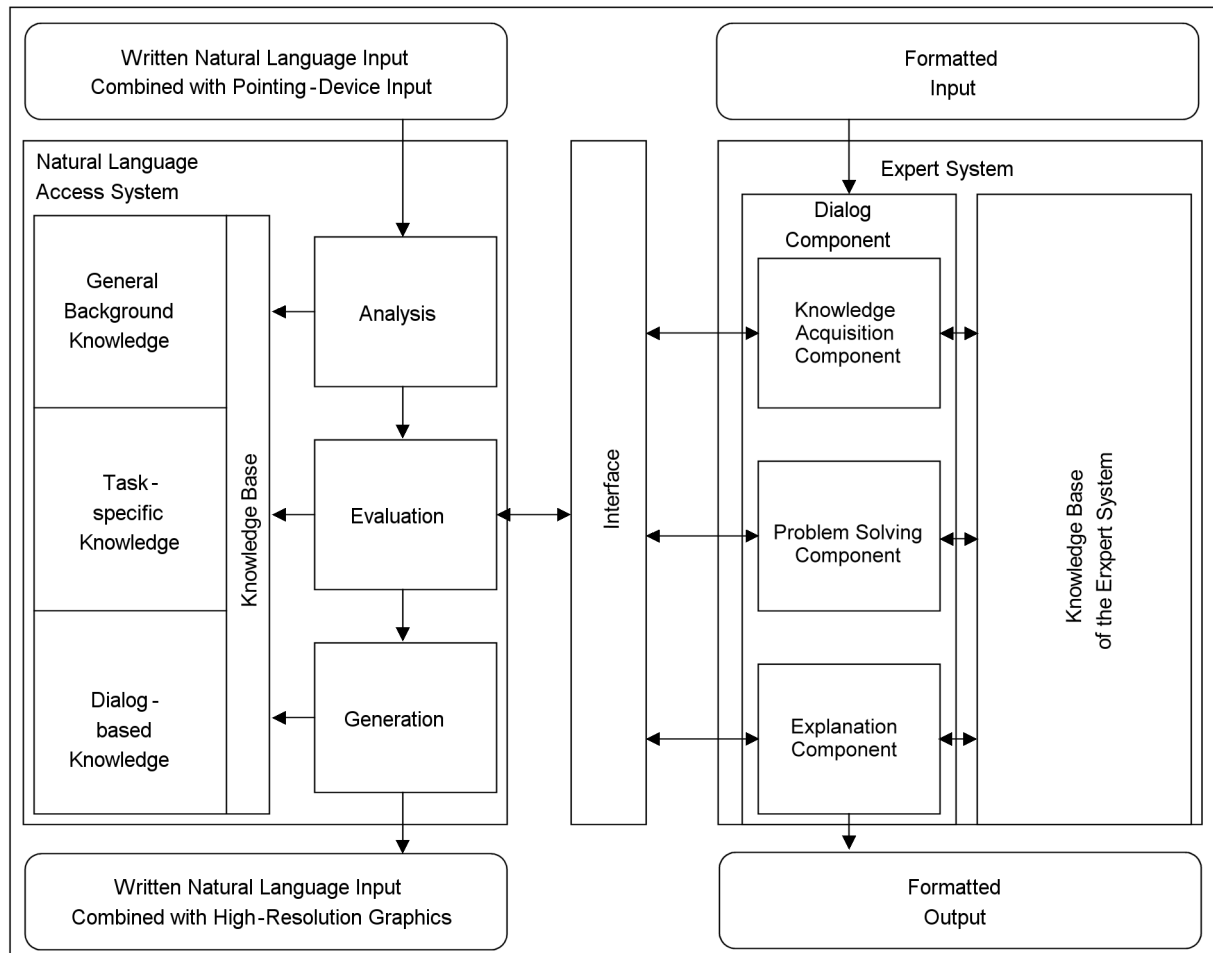


Fig. 4. The Architecture of an NL Interface to Expert Systems.

To give an idea of the sort of dialog we are aiming at, Fig. 6 shows a hypothetical dialog with an interface which provides access to an expert system for fault diagnosis in a technical system. After the dialog sequence (27) - (28), the system must recognize in question (29) that the pronoun 'it' does not refer to the 'large number' just mentioned, but rather to the system disk. The user then asks a why-question (30), and the system provides an explanation like (31). The user's next question (32) presents a difficult problem for the analysis component. It must be recognized as an elliptical why-question which requires an iterative call to the explanation component, so that the system can respond with (33). The user then takes the initiative with (34), whereupon the system immediately recovers the initiative with (35).

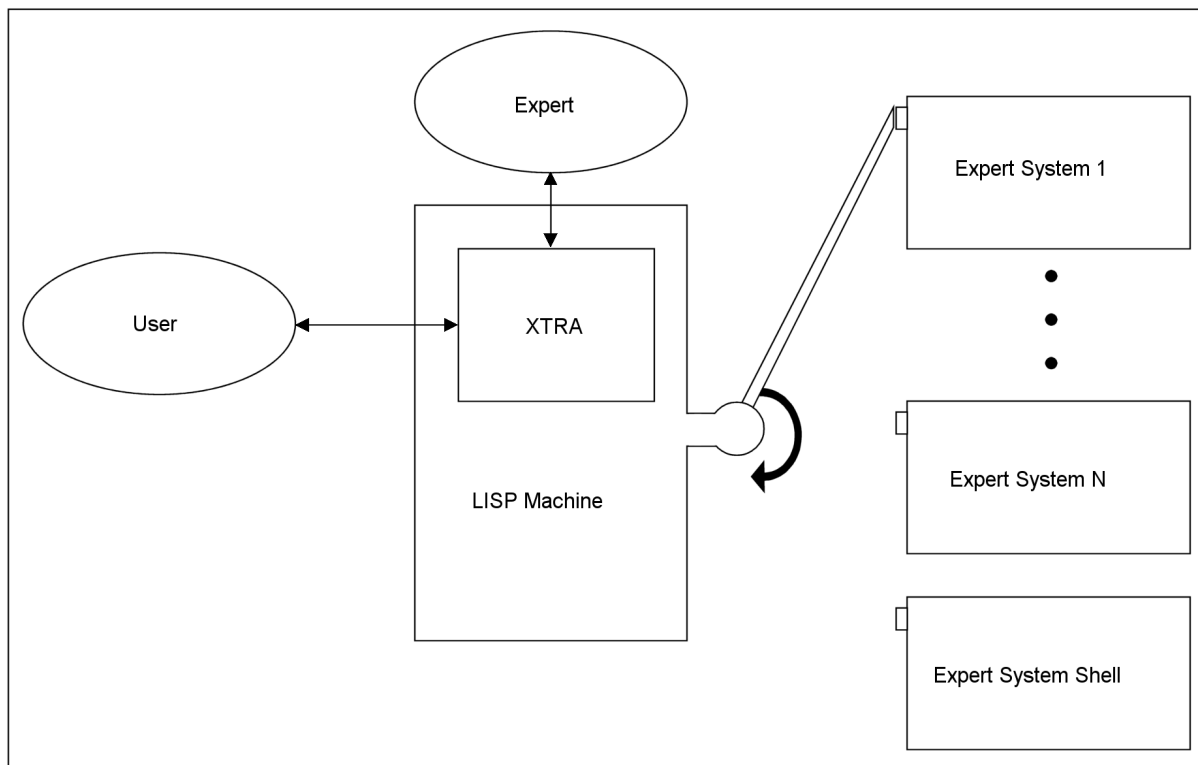


Fig. 5. The Adaptable Access System XTRA.

Aside from expert systems, one of the most important commercial applications for cooperative access system at present is as front-ends to databases. In this case we have a natural language interface which provides access to formatted mass data and a database management system, which handles all accesses and changes to the database. In the access system, knowledge is stored which would normally be possessed by an experienced user of such a database. This knowledge is stored in several different knowledge sources, which are managed efficiently by a Knowledge Base Management System (KBMS).

(...)
 (27) C: Is the system disk reported to be bad?
 (28) U: Yes, a great number of writing errors has been reported.
 (29) C: Is it still write-protected?
 (30) U: Why is the write protection relevant here?
 (31) C: Because some disk tests will be started soon.
 (32) U: Write protection for disc tests?
 (33) C: Yes, because the operating system may be killed if the controller is bad.
 (34) U: Then, I'll start the disk test DISK-1, now.
 (35) C: No, we'll start with DISK-2.
 (...)

Fig. 6. Natural Language Access to an Expert System for Error Detection in Computer Systems.

Technology in this area has already advanced to the point where natural language interfaces to databases can be implemented for personal computers. An example of such natural language interfaces is INTELLECT [5], supplied by Artificial Intelligence Corporation, which runs on an IBM-PC XT and allows the user to query the system in English. This system provides access to the mini-database system dBase II, making use of the graphics package of VISICALC to produce as output a combination of natural language and graphics.

5. Advanced Explanation Components

I would now like to describe briefly some interesting developments in the area of explanation components (cf. also [14]). These components are applied in a wide variety of AI systems, ranging from expert systems and instruction systems to theorem provers, and deductive database systems. A few examples will

- | | |
|------|--|
| (A) | Explanation of inference-based answers: |
| (36) | U: Is the bed fairly hard? |
| (37) | C: I think so. |
| (38) | U: Really? |
| (39) | C: Its mattress is new. |
| (B) | Explanation of conceptual knowledge: |
| (40) | C: There are four seating accommodations. |
| (41) | U: What do you mean, by seating accommodations? |
| (42) | C: A seating, accommodation is a piece of furniture.
The following kinds of seating accommodations are known:
chairs and arm-chairs. ... a seating accommodation has a seat,
usually four legs and often a back-rest. |
| (C) | Explanation of speech acts: |
| (43) | C: May I book the room for you? |
| (44) | U: Why do you ask that? |
| (45) | C: Because I could offer you yet another one. |

Fig. 7. Argumentative Dialog Sequences.

give you an idea of the sorts of explanations which can be produced at present (see Fig. 7). The first type is the explanation of inference-based answers of the system. Suppose, for example, that in the hotel situation the user asks a question like (36), and the system answers 'Yes', and the user then asks 'Really?', the system can explain its answer by a response like (39). What it has done is reveal a premise which it made use of in making the inference which led to the answer. Another type of explanation that is technically possible is the explanation of conceptual knowledge. If, for example, the system at some point during the dialog makes an utterance like (40), and the user would like to know exactly what is meant by 'seating accommodation' (see (41)), it is possible for the system to explicate this concept with a segment of text (see Fig. 7). This text has not been stored in advance but is generated when required on the basis of the system's conceptual knowledge. A capability which has

yet to be implemented is the explanation of speech acts. This will require considerable further research on speech act planning and recognition. For example, if the system asks a question like (43), and the user reacts with (44), the system should be able to supply an answer on a meta-level (see (45)).

I would like to conclude this discussion of explanation components with a description of an intelligent interface to a CAD system. The AI system CADHELP [3] has the task of explaining a complex tool for computer-aided design whose hardware consists of a keyboard, a graphics tablet, and a high-resolution screen (see Fig. 8). CADHELP cooperatively adapts its explanation to the individual user. It may provide an animated graphics display in the upper half of the screen while generating in the lower half a natural language text which explains the various parts of this display. For example, it may state "The lower right-hand side of the screen is the catalog zone."

A particularly striking feature of this project is the fact that both the graphics display and the accompanying text are generated at run time on the basis of a semantic representation. Neither one is stored in advance, so that, e.g., a novice can be given a quite different explanation than an expert. Even in the course of a single dialog, if you ask the system twice about the same point, its second explanation will be very different from its first one.

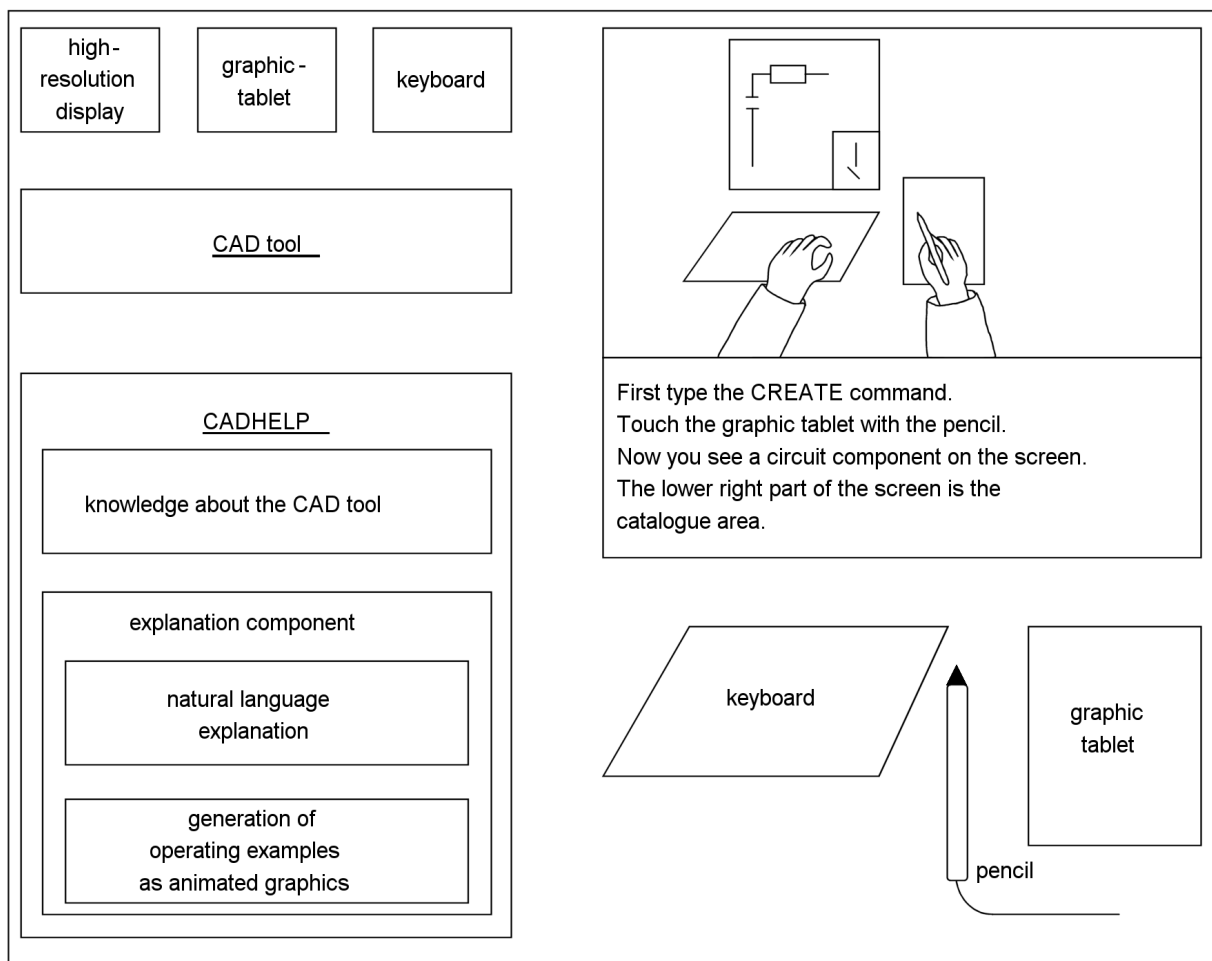


Fig. 8. CADHELP as an Intelligent Interface.

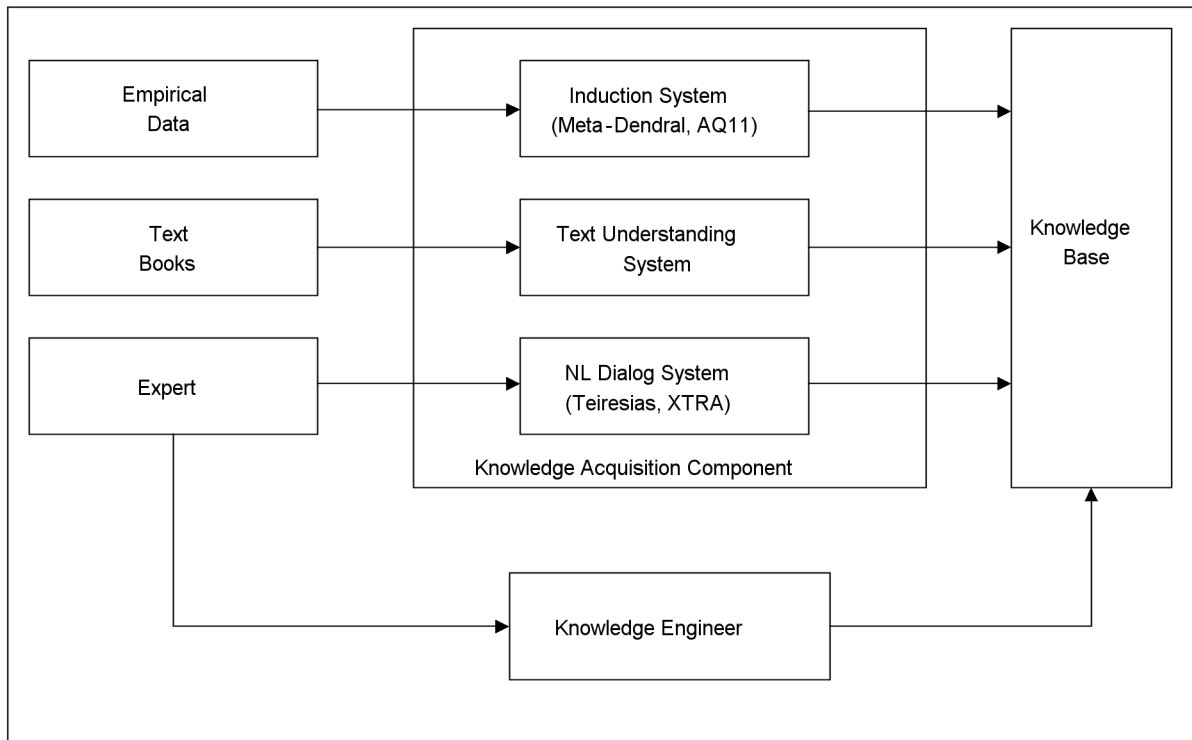


Fig. 9. Modes of Knowledge Acquisition.

6. Concluding Remarks

I would like to conclude my short survey by calling your attention to a new perspective in research on knowledge acquisition (see Fig. 9). Up to now the knowledge base of an AI system has in general been supplied by an expert with the help of an AI specialist. Attempts have been made to transfer at least some of the responsibility for knowledge acquisition to the system itself. The successful attempts have mainly taken the form of a component which elicits knowledge from the expert in dialog.

The system TEIRESIAS [4] is a well-known example of such a component. A more ambitious approach is to have the system extract knowledge from relevant specialist literature, i.e. to use a text understanding system to fill up the knowledge base. At the moment, this is still an unrealistic goal. Given the present state of the art, it is only feasible for a system to understand two or three pages of text on a very narrowly defined subject; it is not possible to process entire books. A still more ambitious approach is to have the system construct the knowledge base itself, applying inductive reasoning to observational data.

Finally, let me emphasize that current NL interfaces are a long way from having broad-based, universal language capabilities comparable to human dialog partners. People claiming that, with the advent of the first commercial NL interfaces such as INTELLECT, the problem of getting computers to understand NL has been solved, do the whole field of AI a serious disservice.

However, the technology for limited NL access systems is available now and there is a large growing market for such cooperative interfaces to a wide variety of software.

A commercial German language access system is presently not available. I am, however, currently coordinating a joint project, funded by the German Ministry for Research and Technology, in which companies like Nixdorf, Siemens and SCS together with academic research groups will develop a prototype of a product for German language access to expert systems within the next three years.

References

- [1] M. Bates, R. Bobrow (1984): Natural Language Interfaces: What's Here, What's Coming and Who Needs it. In: Reitmann, W. (ed.): Artificial Intelligence Applications for Business. Nordwood: Ablex, pp. 179-194.
- [2] J.G. Carbonell, W.M. Boggs, M.L. Mauldin, P.G. Arnik (1983): The XCAL-IBUR Project: A Natural Language Interface to Expert Systems. In: Proc. of IJCAI-83, Karlsruhe, pp. 653-656.
- [3] R.E. Cullingford, M.W. Krueger, M. Selfridge, M.A. Bienkowsky (1982): Automated Explanations as a Component of a Computer-Aided Design System. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 2, pp. 168-181.
- [4] R. Davis (1982): TEIRESIAS: Applications of Meta-level Knowledge. In: Davis, R., Lenat, D.B.: Knowledge-based Systems in Artificial Intelligence. New York: McGraw-Hill, pp. 227-490.
- [5] J. Eisenberg, J. Hill (1984): Using Natural-Language Systems on Personal Computers. In: Byte, January 1984, pp. 226-238.
- [6] T.W. Finin (1983): Providing Help and Advice in Task Oriented Systems. In: Proc. of IJCAI-83, Karlsruhe, pp. 176-178.
- [7] W. Hoepfner, Th. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary, W. Wahlster (1983): Beyond Domain-Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems. In: Proc. of IJCAI-83, Karlsruhe, pp. 588-594.
- [8] A. Jameson, W. Wahlster (1982): User Modeling in Anaphora Generation: Ellipsis and Definite Description. In: Proc. of the first European Conference on Artificial Intelligence, Orsay 1982, pp. 133-138.
- [9] J. King (ed.) (1983): Special Issue on AI and Database (Research. In: SIGART Newsletter, No. 86, October 1983, pp. 32-72.
- [10] E. Mays, A.K. Joshi, B.L. Webber (1982): Taking the Initiative in Natural Language Data, Base Interactions: Monitoring as Response. In: Proc. of the first European Conference on Artificial Intelligence, Orsay, pp. 255-256.
- [11] E. Rich (1983): Users are Individuals: Individualizing User Models. In: Intern. J. Man-Machine Studies 18, pp. 199-214

- [12] L. Siklossy (1978): Impertinent Question-Answering Systems: Justification and Theory. In: Proc. ACM National Conference, pp. 39-44.
- [13] W. Wahlster (1980): Towards a Computational Model for the Semantics of Why-Questions. In: Proc. of the 8th Intern. Conference on Computational Linguistics, COLING-80, Tokyo, pp. 144-150.
- [14] W. Wahlster (1981): Natürlichsprachliche Argumentation in Dialogsystemen. KI-Verfahren zur Rekonstruktion und Erklärung approximativer Inferenzen. Heidelberg: Springer.
- [15] W. Wahlster, H. Marburger, A. Jameson, S. Busemann (1983): Over-Answering Yes-No Questions: Extended Response in a Natural Language Interface to a Vision System. In: Proc. of IJCAI-83, Karlsruhe, pp. 643-646.
- [16] W. Wahlster (1984a): XTRA: Ein natürlichsprachliches Zugangssystem zu Expertensystemen. In: Deussen, P. (ed.): Sonderforschungsbereich Künstliche Intelligenz, Univ. Karlsruhe, pp. 377-402.
- [17] W. Wahlster (1984b): User Models in Dialog Systems. Invited Lecture at COLING-84, to appear in: Computational Linguistics 1984.
- [18] B.L. Webber, T. Finin (1984): In Response: Next Step in Natural Language Interaction. In: Reitman, W. (ed.): Artificial Intelligence Applications for Business. Nordwood: Ablex, pp. 211-234.
- [19] R. Wilensky, Y. Arens, D. Chin (1984): Talking to UNIX in English: An Overview of UC. In: CACM, Vol. 27, No. 7, June 1984, pp. 574-593.