BEYOND DOMAIN-INDEPENDENCE:

Experience with the Development of a German Language Access System To Highly Diverse Background Systems

Wolfgang Hoeppner, Thomas Christaller, Heinz Marburger, Katharina Morik
Bernhard Nebel, Mike O'Leary, Wolfgang Wahlster

Research Unit for Information Science and Artificial Intelligence, University of Hamburg
Mittelweg 179, D-2000 Hamburg 13, F.R. Germany

ABSTRACT

For natural language dialog systems, going beyond domain independence means the attempt to create a core system that can serve as a basis for interfaces to various application classes that differ not only with respect to the domain of discourse but also with respect to dialog type, user type, intended system behavior, and background system. In the design and implementation of HAM-ANS, which is presently operational as an interface to an expert system, a vision system and a data base system, we have shown that going beyond domain independence is possible. HAM-ANS is a large natural language dialog system with both considerable depth and breadth, which accepts typed input in colloquial German and produces typed German responses quickly enough to make it practical for real-time interaction. One goal of this paper is to report on the lessons learned during the realization of the system HAM-ANS. This paper introduces the overall structure of the system's processing units and knowledge sources. In addition we describe some of the innovative features concerning the strategy of semantic interpretation.

I. INTRODUCTION

In the past, considerable effort has been devoted to the development of domain-independent natural language (NL) interfaces. As it turns out, domain independence does not mean that such NL interfaces can be adapted for such diverse information processing systems as CAD, simulating, and operating systems (which we will henceforth call background systems). Thus going beyond domain independence means the attempt to create a core system that can serve as a basis for interfaces to various application classes (see fig. 1) that differ not only with respect to the background system but also with respect to dialog type, user type, intended system behaviour, and discourse domain (see upper half of fig. 1). We define the core system of an NL interface as consisting ideally of those knowledge sources (KSs) and processes that are common to all applications. Of course this is a very ambitious goal that does not seem to be attainable in its full generality in the near future according to the current state of the art, but we have shown that it is feasible in a more limited sense in our design and implementation of the system HAM-ANS (HAMburg Application oriented Natural language System), which is presently operational as an interface to an expert system, a vision system, and a data base system (see lower half of fig. 1).

HAM-ANS is a large NL dialog sytem of both considerable depth and breadth, whose components have been developed over a 3-year period with an effort of approximately 15 man-years. The system accepts typed input in colloquial German and produces typed German responses quickly enough (5-25 seconds on an unloaded PDP-10) to make it practical for real-time interaction. The project represents a massive approach towards developing and implementing an actual application-independent, easily adaptable system. The design of HAM-ANS draws mainly upon work on CO-OP [14], PLANES [24], INTELLECT [8], TEAM [6] and HAM-RPM [7].

Research on HAM-ANS is currently being supported by the German Ministry of Research and Technology (BMFT) under contract 08IT15038

II. HAM-ANS AS AN NL INTERFACE TO HIGHLY DIVERSE BACKGROUND SYSTEMS

Our research strategy has been to build the core system with a set of diverse applications included from the very beginning. As the upper half of figure 1 shows, the three application classes considered in HAM-ANS are defined by choosing a different value for each of the dimensions. Before the capabilities of the working system are demonstrated, let us briefly characterize the communicative situations that we have focused on in the various application classes listed in figure 1.

In the hotel reservation situation the system takes the role of a hotel manager, who tries to persuade the user to book a room (see fig. 2). The caller is assumed to have the overall goal of determining whether the room offered meets his requirements. The system must attempt to recognize the user's specific desires concerning the room as they are revealed - usually indirectly - in his utterances and to make use of the various devices available in natural language that permit the room in question to be presented in a particularly favorable light (e.g. the generation of tendentious descriptions using hedged relative adjectives).

In the traffic domain we presuppose the following situational context: The system is observing a street intersection and supplies the user, who is familiar with the scene but cannot see from his remote location, with information about the traffic at that intersection on the phone (cf. [16]). A geometrical description of the observed scene is provided by the image sequence analysis system NAOS/MORIO [5][19]. In this application, in contrast to the hotel domain, where at the beginning of the dialog the user must be assumed to have no definite knowledge about the interior of the room being offered, both partners are assumed to have shared knowledge about details of the domain discussed. This difference has strongly motivated our design of parameterized components for the generation and resolution of definite descriptions and anaphora.

In the database application [18] HAM-ANS provides marine scientists with NL access to a fishery database, which contains data collected during international expeditions in the South Atlantic and the Indian Ocean. The complex structured data sets, which consist of cruise data concerning the vessels' voyages, krill biology data, relevant net hauls and oceanographic data, are stored in a relational database with a PASCAL/R data base management system. PASCAL/R is an extension of the well-known programming language PASCAL, having the data structure 'relation' and a general content-based selection mechanism for relations [21]. In contrast to the hotel reservation situation, where the system tries to sell a particular room, in accessing the vision system and the database system HAM-ANS has no interest beyond helping the user by responding in a cooperative way.

We believe it to be very important that each new or modified process be testable in an existing and running program, so that each extension of the system takes place under comparable conditions. Only then it is possible to identify early enough in the design process the application-independent components, the interfaces needed between them and application-dependent ones, and the tools that make it feasible to build an NL interface based on the core system. We have attempted to develop the different linguistic, communicative, and cognitive abilities of the core system as evenly as possible.

Often we started with highly non-deterministic but easy to handle control structures for the rapid development of new components. In later stages of development, after careful evaluation of the actual flow of control in these components, optimizations were performed which often produced sequential and simple control structures.

Even in the simplest settings it seems illusory to develop an application-oriented NL interface merely by radically cutting away advanced features from relatively complete, but inefficient NL-systems that are used as vehicles for basic research (cf. [3] [22]). Therefore, in the design and implementation of HAM-ANS we have dealt from the beginning with phenomena like anaphora resolution, user modelling, the generation of extended responses, and focus shifting, always looking for simple and efficient solutions that provide the user with an adequate functionality in limited domains without trying to simulate human performance.

The technical question of communication between HAM-ANS and the background system is not adressed in figure 1. We have implemented a module in HAM-ANS for communicating with interactive background systems according to the ISO reference model for 'Open System Interconnection' [26]. The session control level is realized as an asymmetric interprocess communication; i.e. HAM-ANS controls, for example, the PASCAL/R interpreter as a slave process. This allows the NL interface to proceed with a dialog with the user while PASCAL/R is performing an assigned task. At the presentation level, the background system is expected to be able to transform from external to internal representations and vice versa. This approach seems to contribute to application independence. However, an equivalent module must be defined in the NL interface for each application.

III. AN OVERVIEW OF THE CORE SYSTEM

In HAM-ANS a user's utterance is processed in a multi-level fashion (cf. [9]): Initial and final dialog sections in the hotel reservation application (cf. fig. 2) are handled by a surface-level model which is realized by a script-like dialog-grammar, since these sections of a telephone call consist largely of standard phrases. The user's more specific questions about the hotel

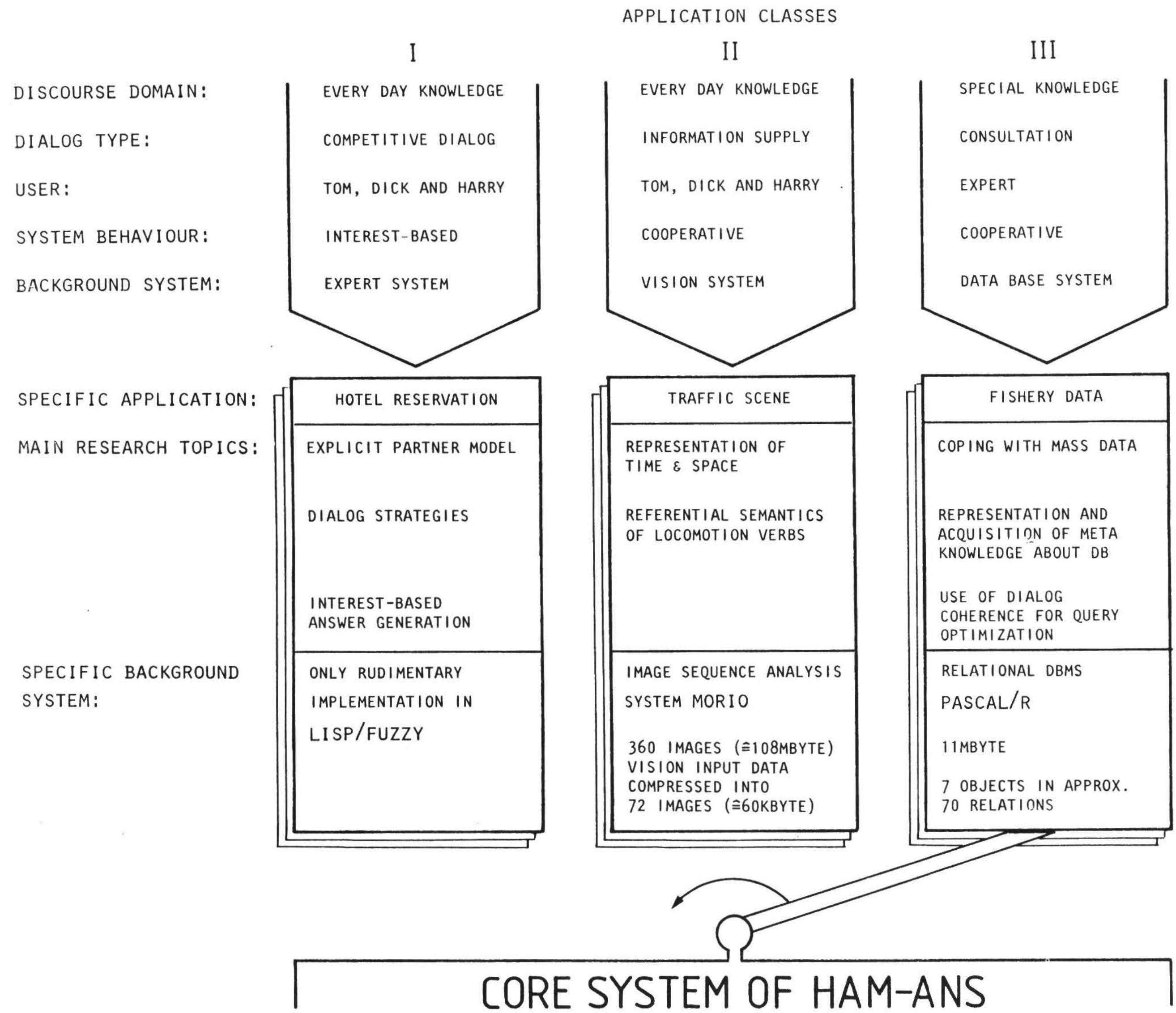

Fig. 1: Some characteristics of the three applications of HAM-ANS

room, however, are analyzed in a linguistically deep manner. In this section, only the second level of processing is considered.

```
HAM-ANS: HOTEL LORELEY, GOOD EVENING!
USER:    Good evening, this is Schmidt,
         University of Hamburg.
HAM-ANS: WHAT CAN I DO FOR YOU?
USER:    I would like a three-bed room
         for two days.
HAM-ANS: SURE, I JUST HAVE A LOOK UP ...
         A ROOM WITH THREE BEDS UNTIL
         JANUARY 26TH?
USER:    That's right.
HAM-ANS: YES, THERE ARE STILL SOME
         VACANCIES.
         IS THERE ANYTHING ELSE YOU
         WOULD LIKE TO KNOW ABOUT
         YOUR ROOM?
.............................................
USER:    Does the room come with a TV
         or radio?
HAM-ANS: YES, THERE IS A RADIO, BUT
         THE ROOM DOESN'T HAVE A TV.
USER:    A telephone?
HAM-ANS: YES.
USER:    Perhaps there's a desk in
         the room?
HAM-ANS: YES, THERE IS IN FACT A DESK.
USER:    Is it a well-lit room?
HAM-ANS: YES.
USER:    Indeed?
HAM-ANS: IT HAS GOT A BIG WINDOW.
USER:    Fine.
.............................................
HAM-ANS: MAY I RESERVE THE ROOM FOR YOU?
USER:    Yes, please.
HAM-ANS: A ROOM WITH THREE BEDS IS BOOKED
         UNDER THE NAME SCHMIDT FROM
         JANUARY 24TH UNTIL 26TH.
         THANK YOU VERY MUCH. GOOD BYE.
USER:    Good bye.
```

Fig. 2: A translated example of a dialog with HAM-ANS in the hotel domain

Figure 3 gives an overview of the most important components in HAM-ANS. Arrows between components and the three major processing units indicate the sequence of control usually followed by the system. As it is impossible to give a detailed description of each processing unit of HAM-ANS in this paper we will focus on the way the system interprets the internal representation of a user's input. The analysis and generation components will only be sketched.

The central representation formalisms of HAM-ANS are the logic-oriented semantic representation languages SURF and DEEP. Although for reasons of efficiency we use a wide range of specialized representation formalisms for the many kinds of knowledge sources needed in HAM-ANS (see fig. 4), it has been very important to have one central meaning representation formalism to facilitate collaboration in the production of such a large-scale NL system, so that code portions generated by all team members work properly together.

SURF and DEEP are designed to be declarative,equivalent to a logic of a higher order, and, because their definition itself is declarative, easily extendable. The difference between the two is that SURF is structurally as close as possible to NL utterances, whereas DEEP is better suited for the evaluation of utterances on the basis of the system's domain specific knowledge. For example, quantifier scope is not yet explicit in SURF representations - quantified substructures are ordered according to their intrinsic scope. This approach is modelled after that of Hendrix [10]. modified for the German quantifier system. SURF is the target language for the parser and the source language for the generator. The components NORMALIZE and INVERSE NORMALIZE transform expressions in SURF into DEEP and DEEP into SURF, respectively.

Some of the salient features of SURF/DEEP are the inclusion of question operators and logical operators for expectability and value-judgement particles such as 'of course' and 'unfortunately' and the representation of vague quantifiers such as 'almost every' and hedged predicates like 'reasonably comfortable'. Moreover SURF/DEEP include constructions for conjoined terms (e.g. 'one bed and several chairs'), noun-phrases with 'of the....' (e.g. 'three to five of these rooms') and complex coordinative conjunctions (e.g. 'but'). SURF and DEEP also use lambda abstraction and meta-predicates for marking deep cases, tense and voice.

Dissimilar applications require that diverse concepts be expressable in the representation languages. For instance, the semantics of locomotion verbs mainly studied in the traffic domain can be adequately represented by case frames instead of term and predicate structures, which serve as a representation formalism for state descriptions occuring typically in the hotel reservation situation. The SURF/DEEP languages are easily complemented by constructs for new concepts and even a complete reformulation of the case system and its integration into the processing components affected by it took less than 5 man-weeks. Thus the design of these central representation languages supports our approach to build a core system for diverse domains.

The processing of a user's NL input starts with a rather elaborate lexical and morphological analysis which reflects the more synthetic character of German as opposed to English. The syntactic analysis comprises two different strategies, both of which use the same ATN-definitions of syntactic categories, e.g. for noun phrases and prepositional phrases. The difference between these strategies lies in the amount of domain knowledge guiding the parsing process. The case-driven analysis strategy [12] contains more specific restrictions for possible case slot fillers than the other strategy, which uses a surface-grammar to cope with word-order variations. The ATN-interpreter incorporates a mechanism to save complete substructures of an utterance as SURF representations, thus enabling the ellipsis recognition component to handle several types of contextual ellipsis by inserting the partial semantic representation into the complete representation of the previous utterance.

The grammar of the core system currently includes rules for complex noun phrases, all common sentence types in various tenses, conjunction (including the most common forms of gapping), replacement ellipsis, all common quantifiers, relative clauses, sentences with sentential complements, and adverbial modifiers. It does not include comparative and superlative constructions, expansion ellipsis, counterfactual sentences or other complex types of subordinate sentences. Before giving a more detailed description of the interpretation unit in the next section, we will sketch the generation of natural language responses (for a detailed description see [23]).

The input to the generation unit is a SURF-representation of a complete utterance. The ellipsis generation component compares this structure to the one constructed by the analysis and identifies the relevant, i.e. new or questioned parts to be passed on to the verbalization component. Possible relevant substructures are rated by applying the system's reconstruction capabilities in an anticipation feedback loop [13].

The task of the verbalization component is to produce a string of canonical words and their grammatical features. This process uses a set of translation rules attached to each category of SURF expressions and is guided by the message to be expressed rather than by the hierarchical structure of the grammar. A special subcomponent provides for the generation of noun phrases as descriptions of domain individuals, which may be part of the SURF expression to be verbalized. Finally the 'surface-transformations' component yields a standard word order schema of the utterance and the correctly inflected forms of the canonical words.

Although, in general, uniformity of representation is one of the architectural principles of AI-systems [20] [25], Davis [4] has pointed out with an example of two larger expert- systems that while a uniform representation is useful for communication between components, the attempt to enforce the same uniformity on all components either fails completely or causes intolerable performance degration. Our experience gained during the development of HAM-ANS also provides evidence against a strong uniformity principle. In an application-oriented NL dialog system there are so many kinds of processes (cf. fig. 3) that require special kinds of knowledge (cf. fig. 4) that a uniform representation schema is unfeasible. There is hardly any common knowledge representation formalism adequate for e.g. the morphological data, conceptual semantic data and case-frame lexicon. Naturally non-uniformity per se is not a design goal in our approach: Wherever possible we have used uniform formalisms (cf. fig. 4, 'implementation formalisms' and [2]).

During the process of developing the lexical analysis component, we tested several representation formalisms. First we used an associative net for representing the word lexicon and DEDUCE procedures (similar to PLANNER's THCONSE theorems) for carrying out morphological analysis [7]. These mechanisms were the same as those used in the evaluation component. The disadvantages were that a blind search was performed by the control regime of the DEDUCE procedures and memory requirements of the associative net were relatively high. We used associativity although it was not needed because of its ease of extendibility. After discovering which operations must be applied in which order the DEDUCE procedures were substituted by a discrimination tree for suffixes and LISP functions to interpret it, freezing the morphological analysis as a strictly sequential deterministic process. Because the lexical analysis component accesses the lexicon only by words, we replaced the associative net by a random access file containing lexical knowledge indexed by words.

Similar experience was gained by using ATNs in the verbalization component (cf. fig. 3). It seems conceptually elegant to use the same representation formalism for generation as for syntactic analysis. Because ATNs only recognize linear structures and not tree structures such as SURF expressions, an ATN must analyze the SURF expression in linear form, thereby implicitly building the structure that is already present in the SURF expression. Having experimented with ATNs we now use the recognizer functions for SURF expressions which are more appropriate in this application.

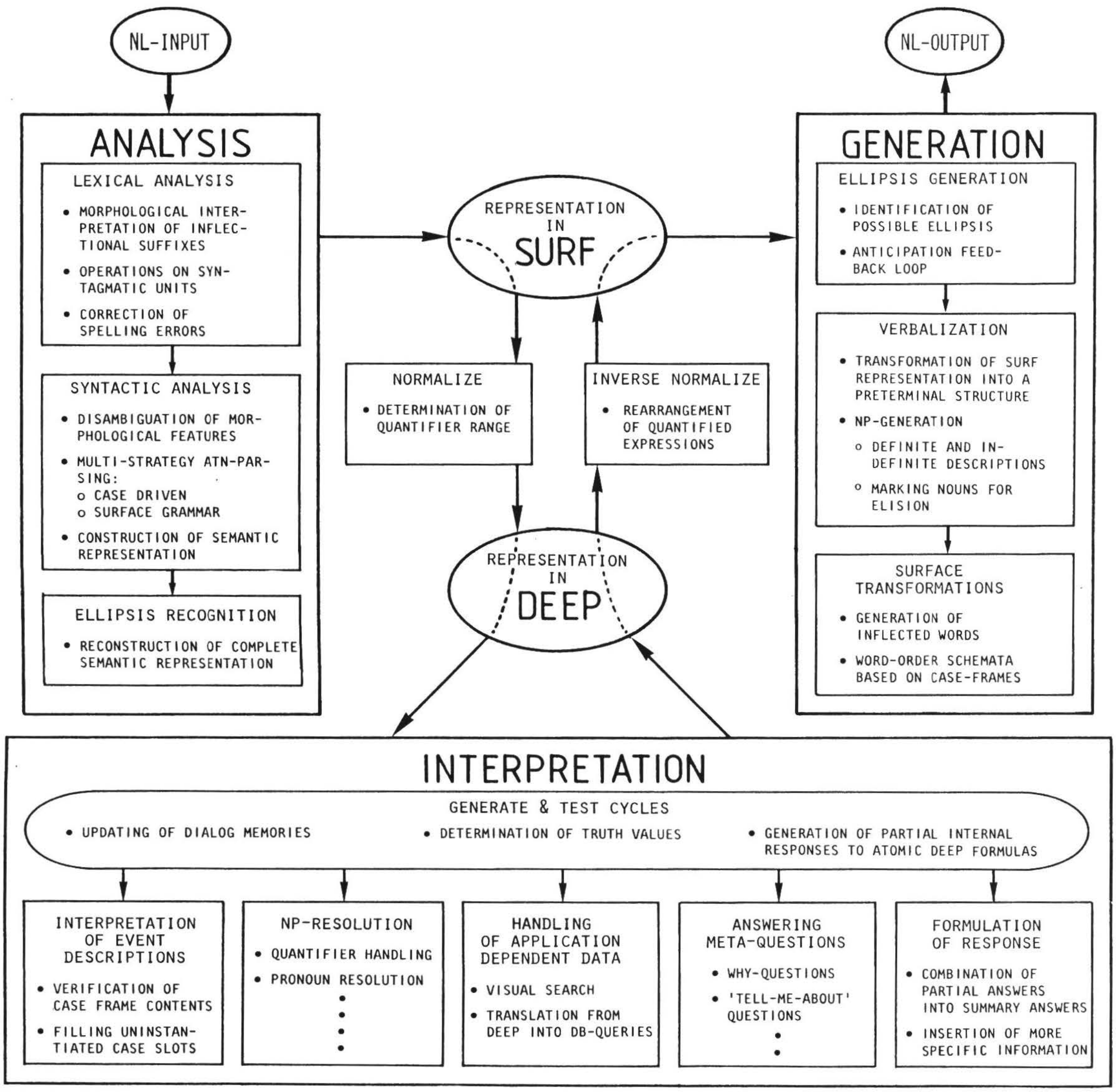

Fig. 3: An overview of the main components of HAM-ANS

KNOWLEDGE SOURCE	NECESSARY FOR EVERY APPLICATION CLASS	NECESSARY FOR EACH APPLICATION INSTANCE OF A CERTAIN CLASS	APPLICATION INDEPENDENT	PERMANENT (p), PERSISTENT BUT CHANGEABLE (c), TEMPORARY (t)	COMPILED OR COMPILABILITY	SIZE IN KBYTE OF THE EXTERNAL REPRESENTATION ON SECONDARY STORAGE	ACCESS: ASSOCIATIVITY (a) VS. ACCESS THROUGH ONE PATH (o), STORAGE IN PRIMARY (p) VS. SECONDARY STORE (WITH DIRECT ACCESS) (s)	IMPLEMENTATION FORMALISM	SPECIAL TOOLS	USE BY COMPONENTS
WORD LEXICON	+	+	(+)	c	-	65	os	PROPERTY LISTS	CONSISTENCY CHECKS, PRESENTATION FACILITIES	LEXICAL ANALYSIS, SURFACE TRANSFORMATION
MORPHOLOGICAL DATA	+	+	+	p	+	38	as	DISCRIMINATION NET	NONE	
SPELLING CORRECTION DATA	+	+	(+)	c	+		op	INDEX TABLE	CONSTRUCTOR	LEXICAL ANALYSIS
PHRASAL LEXICON	+	+	-	p	(+)	11	os	PATTERN ACTION RULES	EDITOR, CONSTRUCTOR	
WORD ORDER RULES	+	+	+	p	-	2.5	op	PATTERN ACTION RULES	NONE	SURFACE TRANSFORMATION
SEMANTIC-ORIENTED GRAMMAR	+	+	+	p	+	45	os	ATN FORMALISM	EDITOR, COMPILER, TRACER, PRETTYPRINTER PRESENTATION FACILITIES	SYNTACTIC ANALYSIS
CASE-FRAME LEXICON	+	+	-	p	(+)	8.1	os	ABSTRACT SURF	NONE	SYNTACTIC ANALYSIS, INTERPRETATION OF EVENT DESCRIPTION
CONCEPTUAL SEMANTIC NET	+	+	(+)	p	+	56	as	ASSOCIATIVE NET	PRESENTATION FACILITIES CONSISTENCY CHECKS	SYNTACTIC ANALYSIS, NP-RESOLUTION, INTERPRETATION OF EVENT DESCRIPTION, VERBALIZATION
REFERENTIAL SEMANTIC NET	-	+	-	p	+	18	as	ASSOCIATIVE NET		
INFERENCE RULES	+	+	(+)	p	-	5.6	ap	DEDUCE PROCEDURES	EDITOR, TRACER, TYPE-CHECK, PRETTYPRINTER	NP-RESOLUTION, INTERPRETATION OF EVENT DESCRIPTION
CONSISTENCY RULES	+	+	+	p	-	7.5	ap	ASSERT PROCEDURES		ANSWERING META-QUESTIONS
SEMANTICS OF ACTION VERBS	+	+	-	p	(+)	49	os	FLAVORS	EDITOR, TRACER, PRETTYPRINTER	INTERPRETATION OF EVENT DESCRIPTION
SCRIPTS	-	-	-	p	(+)	8.1	os	PATTERN ACT. RULE PACKETS	NONE	DIALOG MONITOR
SEMANTIC DIALOGUE MEMORY	+	+	□	t	□	□	ap	SURF	□	ELLIPSIS RECOGNITION, ELLIPSIS GENERATION, NP-RESOLUTION
INFERENCE MEMORY	+	+	□	t	□	□	ap	ASSOCIATIVE NET	□	ANSWERING META-QUESTIONS
RECORD OF DISCOURSE ENTITIES	+	+	□	t	□	□	ap	ASSOCIATIVE NET	□	NP-RESOLUTION, HANDLING OF APPLICATION DEPENDENT DATA, VERBALIZATION
REFERENTIAL NET OF PARTNER	+	+	□	t	□	□	ap	ASSOCIATIVE NET	□	VERBALIZATION
COREFERENCE NETWORK	+	+	□	t	□	□	ap	ASSOCIATIVE NET		
BACKGROUND-SYSTEM-LANGUAGE-SYNTAX	-	+	-	p	-	6.2	os	SYNTAX DEFINITION LANGUAGE	EDITOR, PRESENTATION FACILITIES, CONSTRUCTOR PRETTYPRINTER	HANDLING OF APPLICATION DEPENDENT DATA
DESCRIPTION OF BACKGROUND-SYSTEM OBJECTS	-	+	-	p	-	61	as	ASSOCIATIVE NET	TRANSLATOR FROM DB-SCHEMA	
VISUAL DATA	-	+	-	p	+	25	as	CONTEXTS OF ASSOCIATIVE NETS	EDITOR, TRANSLATOR	
SURF/DEEP-DEFINITION	+	+	+	p	(+)	5.0	os	SYNTAX DEFINITION LANGUAGE	EDITOR, PRESENTATION FACILITIES, CONSTRUCTOR PRETTYPRINTER	IN ALL COMPONENTS EXCEPT LEXICAL ANALYSIS AND SURFACE TRANSFORMATION

□::= not applicable

Fig. 4: An overview of the knowledge sources of HAM-ANS

IV. AN EXAMPLE OF SEMANTIC INTERPRETATION

The interpretation of DEEP expressions may be illustrated by an example taken from a dialog in the hotel-reservation situation. We assume a situation similar to that of the example in figure 2:

User: Is there a lamp next to every bed?
HAM-ANS: Yes, next to almost every one.

Several features are worth noting in the way the system answers this question. First, the system does not merely answer negatively, when a universally quantified expression cannot be verified in a strict sense, but rather tries to modify the quantifier. Second, the system answers yes-no questions in an extended and cooperative way. This is one of the reasons for constructing a system capable of producing complete responses. Third, the system's utterances contain only those parts which bring in new information. In the example above this is particularly notable, since the noun phrase has been reduced down to its quantifier part. We may now take a closer look at the way the example is interpreted and the response is formulated.

The analysis unit constructs a SURF representation reflecting the question of whether the relation NEXT/ TO holds between the subject NP 'a lamp' and the NP 'every bed'. NORMALIZE transforms this SURF formula into a DEEP formula and embeds the first noun phrase into the second one, since the quantifier 'every' is stronger than the indefinite article 'a' (cf. the argument of the first TEST in fig. 5). This DEEP representation excludes the interpretation: 'Does a single lamp exist such that it is next to each bed?'.

The task of interpreting a DEEP formula is governed by a generate and test strategy. Generate and test procedures can be viewed as being activated by pattern-directed invocation and differ from each other in that the generate procedures assign internal object identifiers to variables in DEEP formulas, while the test procedures yield two values, the first of which is either a fully instantiated formula equivalent to the input formula or a modified one depending on the state of affairs, and the second of which indicates the truth value of the input formula in the range [0,1]. In the interpretation phase these two processes interact in such a way that a test attempt activates generate procedures which again call test procedures and so on.

We may now take a closer look at the way the DEEP formula in our example is treated in HAM-ANS. NORMALIZE has ordered the quantified terms for the NPs 'a lamp' and 'every bed' such that the interpretation can proceed from left to right, resolving the NP with the stronger quantifier 'every' before searching for referents for the second NP. This DEEP-formula is shown at the top of figure 5 as the argument of the first TEST. The interpretation process now attempts to verify the complete formula in a top-down manner using the system's inferential capacity and other domain specific knowledge sources (e.g. semantic networks and visual data).

As soon as this process of decomposing the complete formula discovers a structure containing variables, a set of generate procedures is activated and produces the set of object identifiers in the hotel room being considered. Since the client in the initial dialog has asked for a three-bed room, this set is instantiated by (BED1, BED2, BED3). The rest of the formula is then recursively sent to a test process with variables substituted by elements of the reference set for beds one after the other. Being aware that the relation to be verified is a spatial one, a special subset of procedures produces the identifier of the most salient object that is in the relation NEXT/ TO with the specific bed.

A subsequent test phase determines whether the generated object belongs to the class of lamps. If this cannot be shown, as in the example of the deepest TEST attempt, a further object in the neighbourhood of the specific bed is generated and tested until either the conditions are satisfied or no more objects in the relation NEXT/ TO can be generated. Success or failure of the test phases are noted in local memory-registers (HITS and MISSES) which form the basis for composing a complete DEEP formula that comprises the result of the entire interpretation process.

With the quantifier 'every', the interaction between generate and test processes is performed until all of the extensional entities of the concept 'bed' have been inspected. In our example the spatial relation between two of the beds (BED1 and BED3) and a lamp could be established while no object of the class 'lamp' could be generated that met the restriction of NEXT/ TO with respect to BED2. This bulk of specific knowledge could be transferred to the client, but the natural language descriptions necessary to be produced for each of the objects would only be understood by someone with intimate knowledge of the specific hotel room. Therefore the information saved in the local memories is summarized: Similar partial assertions are merged into a single expression, whose quantifier-part indicates the number of successes (or failures).

In addition, the interpretation process uses linguistic hedges to express the relation between successful and unsuccessful partial results. In the example of the three-bed room, only two beds with a lamp next to them are identified and the resulting DEEP quantifier is replaced by the single hedged quantifier 'almost every'. Although the generate and test strategy remains the same in each application class access to external data bases must be able to handle the expressive power of the DB query language.

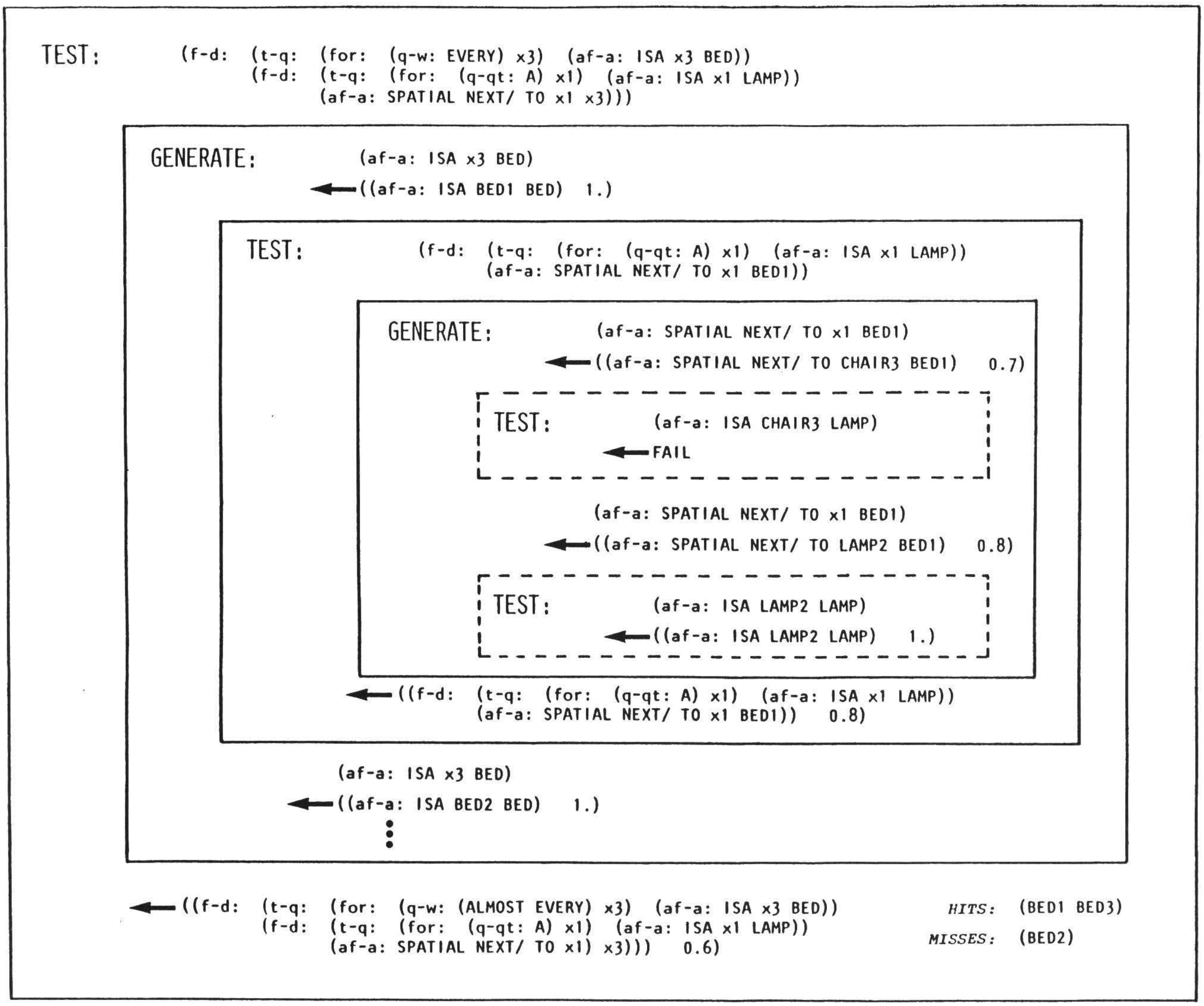

Fig. 5: The evaluation of a DEEP formula in HAM-ANS

V. CONCLUSION

A version of HAM-ANS written in UCI-LISP/FUZZY [17] [15] is presently running under the TOPS10 operating system on a DECsystem 1070 (KI-10). Comprising approximately 900 procedures, the current version requires 200K of 36-bit words.

Building an actual application-independent, easily adaptable system is the research topic in our project HAM-ANS. To meet this demand our research strategy has been to build the core system taking into account a set of highly diverse applications, so that each modification (e.g. extensions to the linguistic, communicative and cognitve capabilities) is testable under varying conditions.

We have provided a detailed presentation of the interpretation process, whereas some of the other distinguishing features of our system, e.g. multi-level processing of user utterances, multi strategy parsing and generation of complete system utterances, were only introduced briefly.

Further work will concentrate on broadening the system's capabilities with a more elaborate partner model, the handling of NL DB updates, facilities for explanation of the systems speech acts, for the treatment of word formation [11], and for NL knowledge acquistion. Although the HAM-ANS core system described is operational, setting our work in relation to the software life cycle, we are still at the system development stage and it will be a long time before the system is ready for distribution to a user community.

REFERENCES

[1] BOLC, L. (ed.): Natural language based computer systems. Hanser/Macmillan, Muenchen, 1980.

[2] CHRISTALLER, Th., v.HAHN, W., HOEPPNER, W., MARBURGER, H. MORIK, K., NEBEL, B., WAHLSTER, W.: Wissensbasierter natuerlichsprachlicher Zugang zu unterschiedlichen Diskursbereichen mit dem KI-System HAM-ANS. In: Slama, R. (ed.): Workshop Sprachverarbeitung, 8. Dezember 1982. GMD Bonn 1982, pp. 100-135.

[3] COHEN, P. R., PERRAULT, C. R., ALLEN, J. F.: Beyond question answering. In: Lehnert, W. G., Ringle, M. H. (eds.): Strategies for natural language processing. Erlbaum, Hillsdale, 1982, pp. 245-274.

[4] DAVIS, R.: Expert systems: Where are we? And where do we go from here? In: The AI Magazine, Vol. 3, No. 2, 1982, pp. 3-22.

[5] DRESCHLER, L., NAGEL, H.-H.: Volumetric model and 3D-trajectory of a moving car derived from monocular TV-frame sequences of a street scene. In: Proc. 7th IJCAI, Vancouver, 1981, pp. 692-697.

[6] GROSZ, B., APPELT, D., ARCHBOLD, A., MOORE, B., HENDRIX, G., HOBBS, J., MARTIN, P., ROBINSON, J., SAGALOWICZ, D., WARREN, D.: TEAM: A transportable natural-language system. SRI International, Technical Note 263, 1982.

[7] v.HAHN, W., HOEPPNER, W., JAMESON, A., WAHLSTER, W.: The anatomy of the natural language dialogue system HAM-RPM. In: Bolc, L., 1980, pp. 119-253.

[8] HARRIS, G. G.: ROBOT: A high performance natural language interface for data base query. In: Bolc, L., 1980, pp. 285-318.

[9] HART, P. E.: Directions for AI in the Eightees. In: SIGART Newsletter, No. 79, 1982, pp. 11-16.

[10] HENDRIX, G. G.: Semantic aspects of translation. In: Walker, D. E. (ed.): Understanding spoken language. North-Holland, New York, 1978, pp. 193-228.

[11] HOEPPNER, W.: A multilayered approach to the handling of word formation. In: Horecky, J. (ed.): COLING-82. Proc. of the 9th Int. Conference on Computational Linguistics, Prague. North-Holland, Amsterdam 1982, pp. 133-138.

[12] HOEPPNER, W.: ATN-Steuerung durch Kasusrahmen. In: Wahlster, W. (ed.): GWAI-82. German Workshop on Artificial Intelligence. Springer, Heidelberg, 1982, pp. 215-226.

[13] JAMESON, A., WAHLSTER, W.: User modelling in anaphora generation: Ellipsis and definite descriptions. In: Proc. ECAI-82, Orsay, 1982, pp. 222-227.

[14] KAPLAN, S. J.: Cooperative responses from a portable natural language data base query system. Ph.D. Dissertation, University of Pennsylvania, Heuristic Programming Project Hpp-79-19, 1979.

[15] LEFAIVRE, R. A.: FUZZY reference manual. Rutgers University, Department of Computer Science, 1977.

[16] MARBURGER, H., NEUMANN, B., NOVAK, H.-J.: Natural language dialogue about moving objects in an automatically analyzed traffic scene. In: Proc. 7th IJCAI, Vancouver, 1981, pp. 49-51.

[17] MEEHAN, J. R.: The new UCI LISP manual. Erlbaum, Hillsdale, 1979.

[18] NEBEL, B., MARBURGER, H.: Das natuerlichsprachliche System HAM-ANS: Intelligenter Zugriff auf heterogene Wissens- und Datenbasen. In: Nehmer, J. (ed.): GI - 12. Jahrestagung. Springer, Heidelberg 1982, pp. 392-402.

[19] NEUMANN, B.: Towards natural language description of real-world image sequences. In: Nehmer, J. (ed.): GI-12. Jahrestagung. Springer, Heidelberg, 1982, pp. 349-358.

[20] PIQUE, J. F., SABATIER, P.: An informative, adaptable and efficient natural language consultable database system. In: Proc. ECAI-82, Orsay, 1982, pp. 250-254.

[21] SCHMIDT, J. W.: Some high level language constructs for data of type relation. In: ACM TODS, Vol. 2, No. 3, 1977, pp. 247-261.

[22] TENNANT, H. R.: Evaluation of natural language processors. Ph.D. Dissertation, Coordinated Science Laboratory, University of Illinois, Report T-103, 1980.

[23] WAHLSTER, W., MARBURGER, H., JAMESON, A., BUSEMANN, St.: Over-answering yes-no questions: extended response in a NL interface to a vision system. In: Proc. 8th IJCAI, Karlsruhe, 1983

[24] WALTZ, D.: An English language question answering system for a large relational data base. In: CACM, Vol. 21, No. 7, 1978, pp. 526-539.

[25] WARREN, D. H. D., PEREIRA, F. C. N.: An efficient easily adaptable system for interpreting natural language queries. Dep. of AI, University of Edinburgh, Research paper No. 155, 1981.

[26] ZIMMERMANN, H.: OSI reference model - the ISO model of architecture for open system interconnection. In: IEEE Transactions on Communication, Vol. COM-28, No. 4, 1980, pp. 425-432.